A CONCURRENCY CONTROL METHOD BASED ON COMMITMENT ORDERING IN MOBILE DATABASES

Ali Karami[1] and Ahmad Baraani-Dastjerdi[2]

[1]Department of Computer Engineering, University of Isfahan, Isfahan, Iran
ali.karami9@yahoo.com
[2] Department of Computer Engineering, University of Isfahan, Isfahan, Iran
ahmadb@eng.ui.ac.ir

ABSTRACT

Disconnection of mobile clients from server, in an unclear time and for an unknown duration, due to mobility of mobile clients, is the most important challenges for concurrency control in mobile database with client-server model. Applying pessimistic common classic methods 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 methods aren`t appropriate in mobile database. In this article, OPCOT concurrency control algorithm is introduced based on optimistic concurrency control method. Reducing communications between mobile client and server, decreasing blocking rate and deadlock of transactions, and increasing concurrency degree are the most important motivation of using optimistic method as the basis method of OPCOT algorithm. To reduce abortion rate of transactions, in execution time of transactions` operators a timestamp is assigned to them. In other to checking commitment ordering property of scheduler, the assigned timestamp is used in server on time of commitment. In this article, serializability of OPCOT algorithm scheduler has been proved by using serializability graph. Results of evaluating simulation show that OPCOT algorithm decreases abortion rate and waiting time of transactions in compare to 2pl and optimistic algorithms.

KEYWORDS

Concurrency Control, Mobile Database, Commitment Ordering

1. INTRODUCTION

In the recent decade, increase processing ability and decrease in prices of mobile devices have concluded in vast use of these devices in mobile environments. Using data processing services in mobile environments like mobile banking, traffic control and e-commerce, and requirements to easy and swift access to information, are among factors leading to advent of mobile database [1, 2].

Relating to communication process and application of mobile devices, client-server, peer to peer, and adhoc architectures are among most recognized architectures of mobile database. In this study, transactions` concurrency control in mobile database is considered by client-server architecture.

Figure 1 displays client-server architecture. Mobile clients and fix host (server) and mobile base stations are three most important elements of this model. In client-server model, mobile clients are connected to fix host through mobile base station [3].

DOI: 10.5121/ijdms.2011.3404

As has been mentioned in [2] Limited energy sources (battery charge), and unexpected disconnection of mobile client from fix host comprise characteristics of mobile devices. These characteristics aren't a specific fault status, but a general and inherent in mobile environment. However, it has been mentioned in [4] that today, using wireless 802.11 protocol and EVDO method, it is possible to have a synchronized and fast connection between mobile client and server. But, characteristics 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.

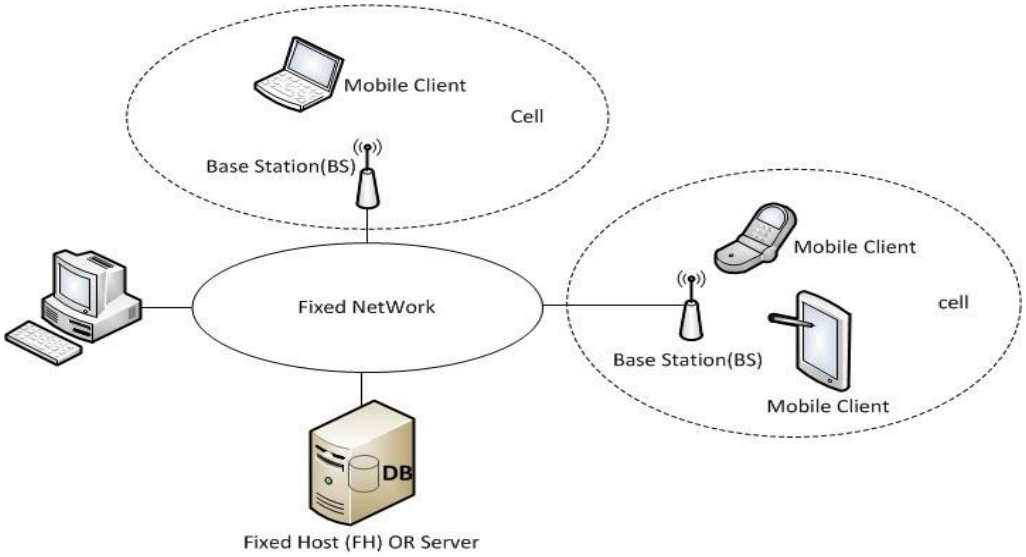

Figure 1. Client-server architecture in mobile database

Concerning aforementioned limitations, concurrency control of concurrent transactions in mobile database face with some difficulties. In pessimistic conventional concurrency protocol based on locking in the time of locking or unlocking, and in protocols based on timestamps, for reading or writing, connection between mobile clients and server is required. In this case non-continuous connection between clients and server led to transactions encounter problems like blocking and long waiting. This leads to inefficiency of pessimistic protocols in mobile environment. In optimistic protocols, no continuous connection between client and server is required. But these protocols are appropriate for optimistic environments in which transaction operators have little conflict. In other non-optimistic environments, using optimistic protocols led to high rate of abortion [2, 3, 5].

In introducing concurrency protocols, issues like reducing rate of abortion, connection between mobile client and server, and time waiting of mobile client, ought to be under considerations. But these measures aren't in accordance in concurrency protocols. In optimistic protocols, communications are less but abortion rate is high; but, in pessimistic ones, the opposite status governs. The main approach in this study is to introduce a concurrency control method based on optimistic concurrency control and commitment ordering schedulers in mobile database. In [6, 7] commitment ordering schedulers for concurrency control has been mentioned.

In the introduced method, just in the beginning and end of transaction, mobile client is connected to server; so, protocols operate optimistically and reduce communications to minimum. Transaction operators and time of operators execution related to time of previous operator execution are added to a list in the transaction execution time. In the time of completing transaction, this list is sent to server; then, based on time of transaction`s operation and the

concept of commitment ordering schedulers, decisions are made regarding abortion or completion of transaction. This method lessens the amount of aborted transactions significantly. Therefore, in this article, by using timestamp of operators and commitment ordering schedulers, the OPCOT protocol is introduced based on optimistic protocol. This can establish a trade-off between rate of abortion and the amount of connection between mobile client and server. OPCOT protocol doesn't possess the problem of deadlock and blocking of transactions.

This article is organized in five sections. Section two is about previously done researches in the field of concurrency control in mobile database. In section three, we consider the introduced method. The evaluation of protocol is mentioned in section four. Final section talks about conclusion and future works.

2. RELATED WORKS

Limitations and specific characteristics of mobile devices cause to little efficiency of conventional concurrency control protocols like 2pl or the other protocols in mobile database. Due to this, various researches have performed in this field. These researches introduce new methods of concurrency control or adapt the existing methods of concurrency control with the requirements of mobile environment.

In section one, it was mentioned that, despite access to fast wireless networks, due to inherent limitations of mobile devices, constant and continuous connection between client and server is not possible through transaction. In locking-based pessimistic protocols, continuous connection of mobile client with server is necessary. If during transaction, this connection will be corrupted, the possibility of unlimited blocking of transactions and also deadlock is appeared. Therefore, based on time out, some researchers have been done regarding methods of non-exclusive locks. In these methods, if transaction doesn't end within the estimated time, locks get caught by transaction will release. References [8, 9] are methods of non-exclusive locks based on time out. In reference [9], a dynamic timer was used to solve the problem of transactions` blocking. In this method, transactions should be finished in specified time, otherwise they would abort. In reference [10] for increasing the efficiency of [9], 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. Also, in non-exclusive methods, there is no priority in the order transactions; that is, due to wrong estimation of the required time, a short and less important transaction gets the place of a long and high prioritized one.

A protocol based on AVI is proposed in [11]. This method has still the problem of transaction blocking and computational overhead. This problem has been mentioned in reference [12]. Multi-version concurrency control based on MV2PL protocol and timestamp being introduced in [13] are in fact an extension of method [14].

Hybrid concurrency control protocol of [15] is consisted of optimistic concurrency control protocol and the method based on lock. In this method, it's been attempted to reduce the abortion rate of transactions in optimistic concurrency control by applying weak locking mechanism. But, the number of messages communicates between mobile client and server is high.

In [16, 17], combination of optimistic and pessimistic is performed according to the semantic of operators. In [16], 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 [12] 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.

Usually, in wireless networks, bandwidth from mobile client to server is less than the bandwidth from server to mobile client. Reference [18] Puts emphasis on asymmetry of connecting bandwidth between mobile client and server. In [18] 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 [19] is adapted for broadcast environments. Also the introduced method in [20] is suitable for broadcastenvironments.

Increasing concurrency degree, decreasing abortion rate, time waiting, and amount of connection between mobile client and server, lack of deadlock and starvation, reducing computational overhead and communicated data among client and server are among objectives to be considered in concurrency control protocol. Because of the opposite direction of these objectives, in conducted researches, some parameters are optimized. Researches in the field of optimistic concurrency control aim to lessen high rate of transactions` abortion in mobile environment. Pessimistic methods are introduced with the aim of reducing connection between mobile client and server and also transactions blocking in mobile environment.

Considering above purposes, in this study, OPCOT concurrency control method is introduced to increase the concurrency degree of transactions and lessen waiting time of transactions and amount of connection between server and mobile client.

3. THE PROPOSED METHOD

The main purpose of this study is to decrease abortion rate of transactions with minimum connection of mobile client and server, and decrease waiting time of mobile clients. To do this, OPCOT algorithm is introduced by assigned timestamp to operators and based on optimistic method according to concept of commitment ordering schedulers.

Transactions are done locally; then, they are sent to server for final stage. In this method, relative timestamps are assigned to operators. Finally, operators` list containing timestamps is transferred to server. The absolute time of operators` execution is calculated by using relative timestamps in server. This is done to ignore effect of local time of mobile clients and synchronize operator timestamps as globally with server time. In server, if the scheduler of transaction is commit order, then transaction would commit successfully.

4. EVALUATION OF THE INTRODUCED METHOD

In this section, the method of concurrency control discussed in this study would be evaluated. To do this, it is compared with concurrency control methods of optimistic and 2PL. Because OPCOT protocol is based on optimistic methods hence optimistic concurrency control protocol is selected for comparison and evaluation of introduced protocol. Indeed OPCOT protocol is based on optimistic methods that use commitment ordering scheduler with timestamp technique for transaction operators. Also the proposed protocol is compared with pessimistic protocols (2PL).

In order to evaluate and compare performance of OPCOT, optimistic and 2PL these protocols will be simulated. According to our knowledge, there aren't any simulators for concurrency control simulation in mobile databases. Due to effect of mobile clients movement on concurrency control, selected simulator must support mobile networks and also be able to implement user defined applications and protocols. We use open source software of network simulator NS-3 [21, 22]. This software has the ability of defining new protocol in various network layers, and also supports implementation of mobile networks.

For simulation, we use NS-3 version 3.11 that was installed on Linux Ubuntu 11.04 operating system in dell laptop with Intel(R) Core(TM) 2 Duo CPU 2.00GHz and 4 GB of memory.

Simulation Can be done using Python or C + + languages in the NS3 simulator. In order to add user defined protocols in NS3 layers, protocols were designed as classes. These classes must be inherited from related classes in corresponding layers and related function must be overwritten [21, 22]. Hence, the considered protocols, after implementing by C++ language, were added to NS3 as application layer protocols. For example we implemented a class for OPCOT protocol in NS3 application layer. This class inherited from NS3 application class.

After adding intended protocols to NS3 simulator, mobile networks simulation with the following characteristics was done. A square-shaped area with 2000×2000 meters space, a fix host (server), 10 mobile base stations, and 500 mobile clients are the characteristics of simulated mobile network. Also implemented protocols that were added to NS3 are used by clients and server. Clients move randomly in different directions and communicate their requirements by mobile base station to the server.

Read and write instructions (with R(X) and W(X) notation in this paper) are Instructions which were used in the transaction. The number of instructions in each transaction, which is performed by mobile clients, is considered as floor of normal random variable with mean of 50 instructions and variance of 10 N (50, 10). The amount of reading and writing orders is instructions equal in each transaction.

In server, a table with 1000 data entity was considered. Also, to evaluate the effect of conflict rate in evaluation parameters, in addition to simulation with 1000 data entity, above mentioned mobile network was simulated with 10000 data entity in server. With the same transaction when the number of data entity in server was increased, the conflict rate was decreased, because probability of using same data with concurrent transactions will be reduced.

For each evaluated method, 100 runs with various numbers of transactions were done. Results of simulation are presented in the following pages as plot. For better evaluation, by using regression model, an approximate function is fitted to points. The R^2 coefficient of each fitted model is greater than %80. Plots and regression models were created by R. R is open source statistical analysis software. reference [23] is a quick guide about R software.

5. CONCLUSION AND FUTURE WORKS

In this article, OPCOT concurrency control based on optimistic method and commitment ordering schedulers in mobile database 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, OPCOT algorithm, based on commitment ordering schedulers, applies the technique of operators` time stamping. This leads to increase in the rate of commitment of transactions. Serializability of scheduler in introduced method was approved. The main problem of OPCOT method is overhead of timestamps and their calculations. This overhead is worthless and is tolerable. We plan to improve this method by introducing it in multi-version status.

.