International Journal of Database Theory and Application
Vol.8, No.1 (2015), pp.205-212
http://dx.doi.org/10.14257/ijdta.2015.8.1.21

A Heuristic approach for Effective Management of Concurrent Transactions in Mobile Environments

Salman Abdul Moiz

School of Computer & Information Sciences, University of Hyderabad
Salman.abdul.moiz@gmail.com

Abstract

The inherent limitations of mobile environments like disconnections, mobility, bandwidth etc affects the performance of system when several mobile clients access the shared data items at the same time. Several timeout based strategies are implemented in literature to deal with the starvation of data items during disconnections and mobility. This increases the abort and rollback rates. In the proposed heuristic approach, the time for execution of a transaction can be predicted based on the given timer value for a specific mobile client. The simulation results show the time for execution can be dynamically estimated given the timer value and vice versa for a particular request originating from specific mobile clients. This increases the throughput of the system as compared to the timer based strategies proposed in the literature.

Keywords: *Concurrency, mobile clients, heuristics, timer, time for execution*

1. Introduction

When multiple mobile clients access the shared data items simultaneously it leads to data conflicts resulting in inconsistency of data items. Due to distributed nature of mobile transactions and the inherent limitations in processing of transactions on move makes concurrency control a challenging problem.

Traditionally concurrency is guaranteed by locking the shared data items. However due to disconnections and mobility, it may lead to starvation of shared resources. The timer based strategies addressed this problem. However this decreases the throughput of the system.

In this paper a heuristic approach is proposed which evaluates the time for execution of a transaction based on the given timer value for specific mobile clients. The intuition is that the time taken for executing a transaction differs from one mobile client to the other. Further for a specified timer value by the service provider the proposed strategy can evaluate the time taken to execute the transaction using regression analysis. The simulation results show better throughput in addition to increase in commit rate of the transactions.

The paper is organized as follows: Section -2 describes the related work; Section -3 presents the architecture of mobile database system. Section-4 presents the heuristic approach that estimates the expected time for execution for a given timer value & mobile client. Section -5 presents the simulated results and Section -6 concludes the paper.

2. Related Work

When multiple mobile hosts access the same date items simultaneously then it may lead to data inconsistency. One of the primary problems in mobile data base environment is Data Conflict anomaly [1]. Optimistic concurrency control techniques are proposed to resolve the data conflict in transaction validation phase. However it may lead to delay in responding to the request. Two phase locking protocol [2] guarantees serializability in

ISSN: 2005-4270 IJDTA

transaction processing but it requires mobile host to remain continuously connected with server. DHP-2PL [3] is a locking protocol based on high priority two phase locking to achieve concurrency control in mobile environments. However in DHP-2PL the lower priority transaction may restart again and again due to several restarts of lower priority transactions. Transaction Commit on Timeout Protocol (TCOT) [4] works on timeout principle. However whenever the timer is to be updated the request need to be resubmitted to the fixed host. Epsilon serializability [5] is used for achieving consistency during concurrent access. However the broadcasting of data objects may hinder the performance of the system in presence of weak connectivity and disconnections.

In the proposed approach, attempt is made to estimate time for execution of transaction given a timer value so as to increase the throughput of the system. The timer value may also be estimated using the execution time of successful transactions whose log is available at the mobile middleware.

3. Mobile Database Architecture

In mobile database environment, the transaction request is initiated at mobile clients or mobile host but the mode of execution of transaction varies. The transaction may be totally executed on fixed host or the transaction is totally executed at mobile hosts or the execution of transaction is shared among mobile host and fixed host respectively.

The Figure 1 describes the generic architecture of the mobile database system. It is based on the three tier architecture. The mobile hosts or mobile clients are devices which can move from one place to another. The mobile hosts or mobile clients may request for transaction on move from any place and any device.

Fixed hosts are stationary servers which can store the database. The base station acts as an interface between mobile host and fixed host respectively. Mobile host is connected to base station through a wireless link and the base station is connected to a fixed host in a wired mode.

Mobile middleware acts as an interface between mobile host and fixed host respectively. Since the there could be various databases at the fixed host for varied applications, the mobile middleware realizes the transaction management. The request for execution of transaction is initiated at mobile host which is sent to the mobile middleware. The mobile middleware fetches the required database fragment needed to execute the transaction and sends it to the mobile host.

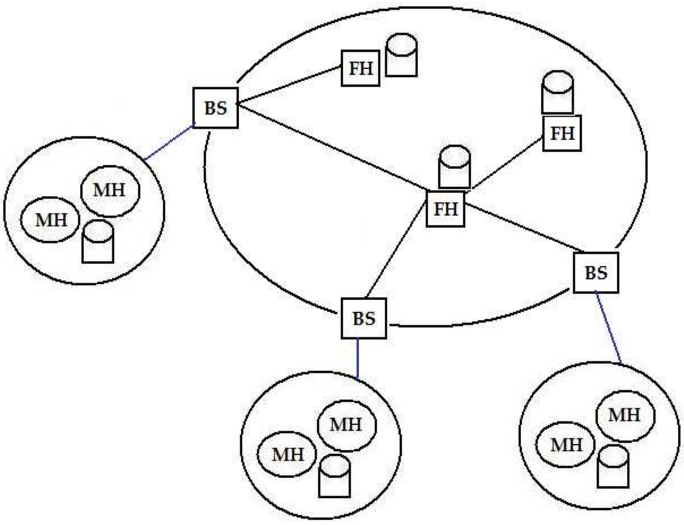


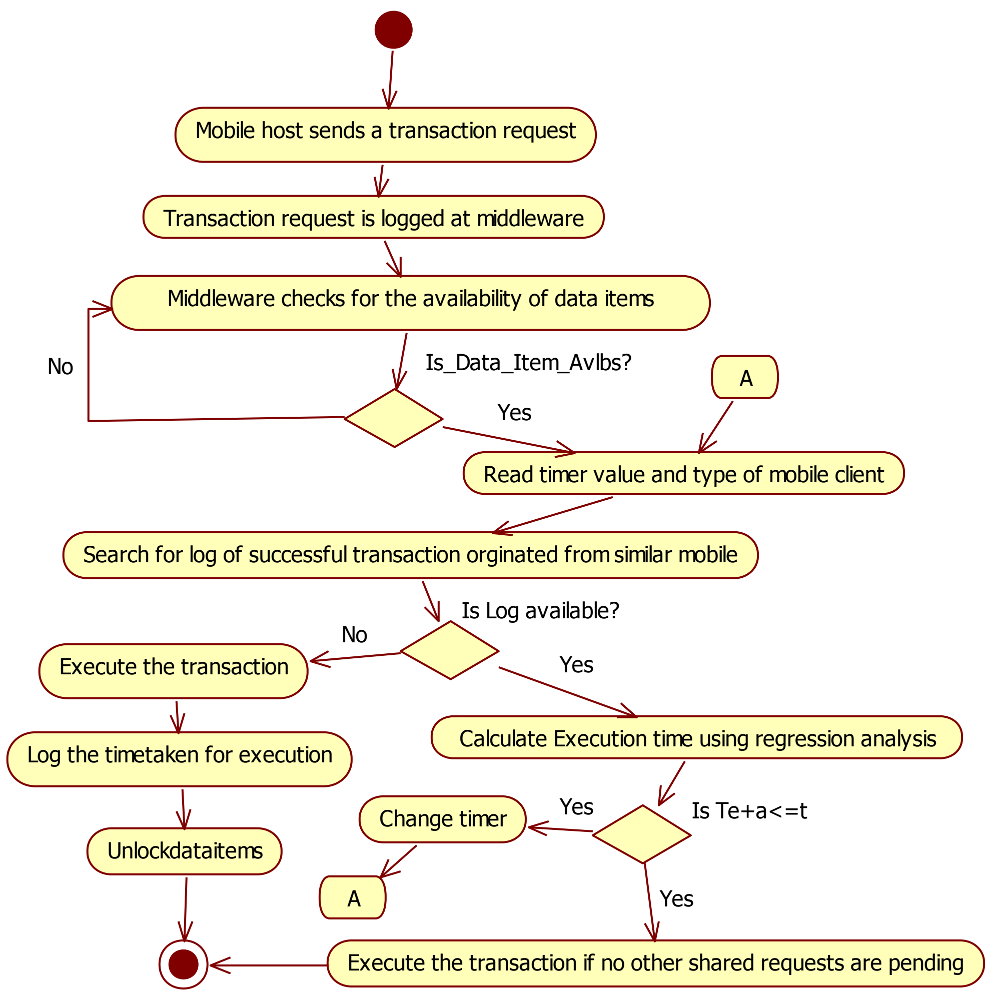
Figure 1. Mobile Database Architecture [6, 7]

The mobile host starts executing the transaction locally. If the mobile host is disconnected or when it moves from one cell to another the transaction can execute locally. Once the transaction is successfully executed the results are integrated with the server to get the system in the consistent state.

4. Heuristic Approach for Predicting Execution Time

When multiple clients request to access the shared data items, the data items are locked to avoid any conflicts. To avoid locking for invariant time period, each transaction is expected to be completed in time "t" (timer). If the transaction doesn't return the result of the execution within the time period "t" it was either rolled back or aborted. However if the transaction is not completed within the timer "T" (threshold limit), it was aborted.

When log of several successful transactions exist it is possible to predict the time for execution using several clustering algorithms [x]. These algorithms help in estimating the time for execution initiated by particular type of mobile client for a specific transaction Ti. If the transaction log specifies the timer value (t) and time taken to execute the transaction (te), then the proposed heuristic approach predicts the expected time for execution for a given timer and type of mobile client pair. This helps in knowing the time taken by transaction under different timer options available and subsequently the intelligent agent can schedule the transaction which increases the throughput of the system. The proposed heuristic strategy is depicted in Figure 2.



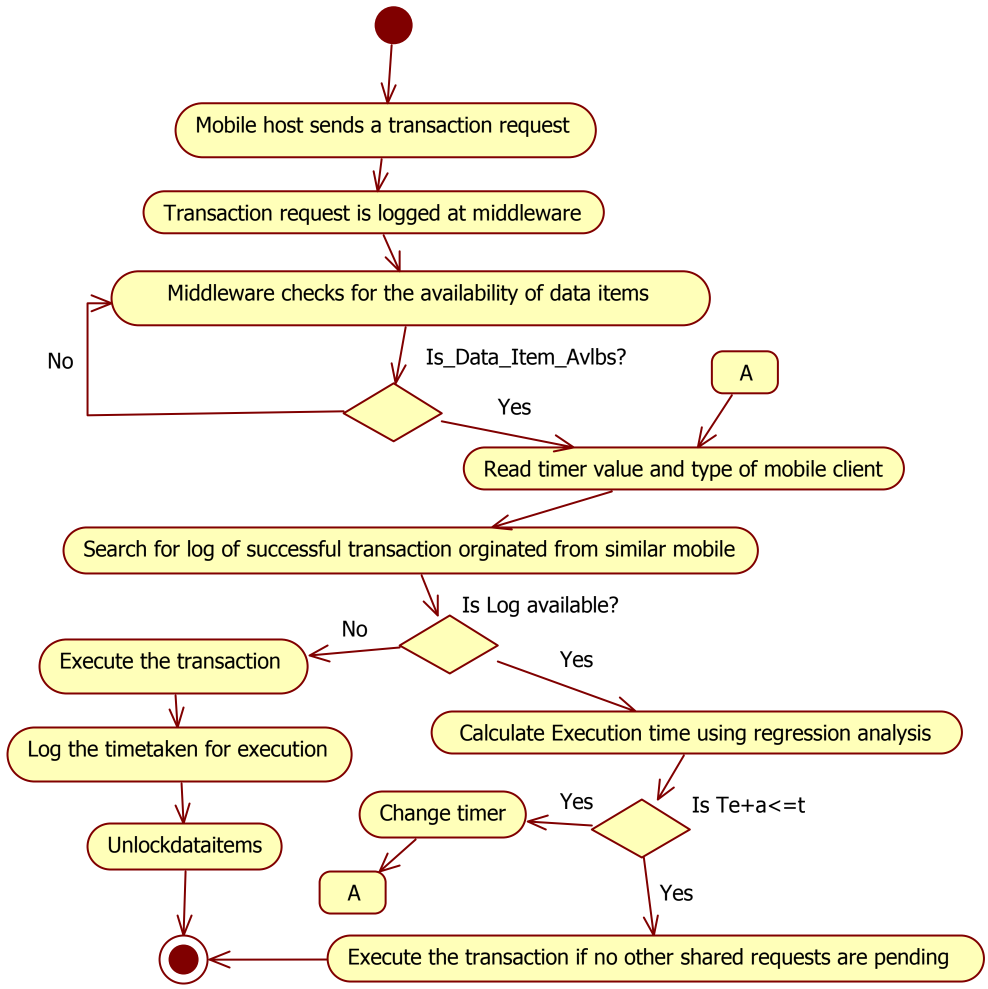


Figure 2. Predicting Time for Execution using Regression Analysis

The request for transaction is initiated by mobile clients by accessing the mobile application installed on it. The request is received by the mobile middleware which checks for the availability data items. If the data items are locked by other mobile clients, the request remains in the waiting queue.

If the data item(s) for requested transaction is/are available then the service provider enters the timer value to predict the time for execution. The time for execution for a given transaction may vary for different scenarios. The time for execution for same transaction varies from one mobile client to other depending on the features of mobile clients on which the transaction executes. Smart devices may execute the transaction in less time as compared to other devices. The time for execution for a given transaction for similar mobile clients may vary depending on the timer value.

The repository is expected to contain the triplet [Mobile_client_type, Timer(t), Time of execution(Te)]. Given this triplet the time for execution can be predicted for a given timer value (t) for the particular mobile client. However if no such log exist then the transaction is executed in a normal fashion and its time for execution is stored in the log. When many such future transactions executes from similar mobile clients the time for execution taken at each instance can be used to predict the expected time for execution.

4.1. Regression Analysis for Expected Time of Execution.

A function is said to be linear when pairs of X, Y values can be plotted as a straight line. If there are two variables X and Y, we shall have two regression lines. One is regression line of Y on X and the other is regression line of X on Y. The regression line of Y on X gives the most probable value of Y for a given value of X. Similarly the regression line of X on Y gives the most probable value of X for a given value of Y.

In the proposed approach X is the timer "t" and Y is expected time for execution (Te). The first regression line equation helps in predicting the expected time for execution when a given timer "t" is specified for a specific mobile client. The second regression line equation is used for predicting the timer values when time for executions is specified.

In certain situation the time for execution (Te) may still be greater than the timer (t). This happens when the timer expires but no other pending transactions requests for same shared data items. In this situation the transaction continues to execute even though the time taken to complete the execution is more than the timer value.

Regression equation of Time of execution (Y) on specified timer (X):

$$\sum Y = na + b\sum X$$

$$\sum XY = a\sum X + b\sum X^2$$

Regression equation of Timer (X) on Time of execution (Y)

$$\sum X = na + b\sum Y$$

$\sum XY = a\sum Y + b\sum Y^2$. Where $\sum X$ is the sum of X_i's

5. Results

The proposed heuristic approach for managing concurrent transactions is simulated for a banking application from varied mobile devices. The mobile client types are characterized by the specification of mobile devices. The log of few of the transactions is specified in Table 1.

Table 1. Log of Transactions

Mobile_Client_Type	Transaction	Timer (t)	Time for Execution(Te)
M1	T1	4	3
M2	T2	5	5
M1	T1	5	5
M1	T1	5	4
M2	T2	5	6
M1	T2	6	6
M2	T2	5	4
M2	T1	6	5

Whenever a request for a transaction (say T1) originates from mobile device type (say M1), then a subset of log is created as follows. The subset of log for Transaction T1 originating from mobile device M1 is specified in Table 2.

Table 2. Log of Transaction T1 Originating from Mobile Client Type M1

Timer (X)	Te(Y)	X^2	Y^2	XY
4	3	16	9	12
5	5	25	25	25
5	4	25	16	20
$\sum X=14$	$\sum Y=12$	$\sum X^2=66$	$\sum Y^2=50$	$\sum XY=57$

Regression equation of Time of execution (Y) on specified timer (X):

$3a + 14b = 12$

$14a + 66b = 57$

$Y = -3 + 3X/2$

For any given timer X the expected time for execution Y is calculated for the Mobile M1 requesting for transaction T1. This helps in deciding whether to proceed with the current transaction or to schedule the next transaction. This increases the throughput of the system.

Compared to the previous approach the throughput of the system has considerably increased without the change in the commit rate. The simulation results also show that the waiting time for transactions is considerably reduced as compared to approaches proposed in literature.

Similarly Regression equation of Timer (X) on Time of execution (Y) is

$3a + 12b = 14$

$12a + 50b = 57$

$X = 20/3 - Y/2$

For any given expected time for execution Y the timer value X is calculated for the Mobile M1 requesting for transaction T1. This helps in deciding the timer value when the log of time for executions is known. This approach solves the problem of setting the timer value. Initially the service provider may set a random timer value “t”. Once a log of successful transactions is available the regression helps in estimating dynamically the timer value for a particular instance.

6. Conclusion

Concurrency control is one of the important building blocks of transaction management. As the device requesting a particular transaction may be disconnected after blocking the resources, there is a need to deal with concurrency by increasing the throughput of the system. In the proposed approach the time for execution of a transaction can be predicted for a specified timer value. Given a log of execution time of successful transaction, the timer which is to be set to increase the throughput of the system can be predicted. This increases the performance of the system. It also reduces the waiting time considerably as the rollback operations are mostly reduced.

Acknowledgements

This paper is a revised and expanded version of a paper entitled “Improving the Success rate of Concurrent Mobile Transactions by predicting Time for Execution” presented at UNESST 2014”, Hainan, China during December 20-23, 2014.

References

[1] V. C.s Lee, "Efficient Validation of Mobile Transactions in Wireless Environments", Journal of System Software, vol. 69, no. 1-2, **(2004)**, pp. 183-193.
[2] A. A. Alivey, "Two Phase Locking Protocol in Distributed Database", Proceedings of IMM Of NAS of Azerbaijan, **(2003)**, pp. 251-254.
[3] K. Yiu Lam, "Concurrency Control in Real Time Database Systems", Information Systems, vol. 25, No. 4, pp. 261-286, 2000.
[4] V. Kumar, "A Timeout based Mobile Transaction Commitment Protocol", IEEE Transaction of Computers, vol. 51, no.10, **(2002)**.
[5] N. Prabhu and V. Kumar, "Concurrency Control in Mobile Database System", 18^{th} International Conference on Advanced Information Network & Application (AINA04), **(2004)**, pp. 83-86.
[6] P. Serranao Alvarado, "A Survey of Mobile Transactions", Distributed & Parallel Database, Kluwer Academic Publishers, vol. 16, **(2004)**, pp. 193-230.
[7] S. Abdul Moiz, "Concurrency Control in Mobile Environments: Issues & Challenges", International Journal of Database Management System, vol.3, no. 4, pp. 147-159, **(2011)**.

Author

Salman Abdul Moiz, is working as Associate Professor in School of Computer & Information Sciences, University of Hyderabad. He did his MCA, M.Tech(CSE), M.Phil(CS) and P.hD (CSE). His area of interest includes Mobile Databases, Software reusability & Security. He is a fellow of IETE & member of IEEE, ACM, CSI, IAENG, ISRS & EWB.