DISCONNECTION TOLERANCE IN RELIABLE TIMEOUT BASED COMMIT PROTOCOL

Bharati Harsoor[1], Dr. S.Ramachandram[2]

[1]Dept of CSE, University College of Engg,
Osmania University, Hyderabad, India
bharati_a@rediffmail.com
[2] Professor, Dept of CSE, University College of Engg,
Osmania University, Hyderabad, India
schandram@gmail.com

ABSTRACT

The transaction processing on the mobile network may lead to disconnection due to poor bandwidth. To deploy traditional database applications into the mobile networks the transactions are to be executed consistently without hindrance of blocking. In this paper, the proposal is made to handle the transactions efficiently despite the frequent disconnections of the wireless network. With the use of log at MH & also at FH, it is possible to process transactions locally during disconnection & reflect updates after getting reconnected. This may lead to non-blocking situations, reducing the message costs, & enhancing the throughput. The simulation results specify the performance analysis metrics.

KEYWORDS

Mobile Transactions, Flash memory, Transaction Log, Transaction Recovery, Network disconnection.

1. INTRODUCTION

The mobile computing environment consisting of fixed host (FH) and mobile host (MH). While in motion, a MH can communicate through a wireless interface supported by some FHs that act as Base Station (BS)—also called as MSS. Compared to traditional ones, wireless networks have particular characteristics like low and variable bandwidth [5]. These characteristic makes the transmission cost more expensive & bandwidth consumption becomes an important concern.

The user issues transactions from his/her mobile unit (MU), the transaction may not be completely executed at the MU so it is fragmented and distributed among database servers for execution. The MU receives results based on the timestamps. In this application, each mobile device contains a view of the database and it must be possible for each user to reach the fixed site in order to execute a transaction. The results must be provided by the server as timely and possible despite the network disconnections. The main problem is then to insure transactions computing continuity. This creates distributed mode of execution. The transactions executed in distributed networks often require an atomic execution. Guaranteeing atomicity in mobile networks involves a lot more challenges than in fixed-wired networks.

These challenges mostly concern network failures, e.g. network partitioning and node disconnection, each of which involves the risk of infinite blocking and can lead to a high number of aborts. With this mode of execution it is difficult to enforce ACID properties in

DOI : 10.5121/ijasuc.2010.1407

mobile transactions. In this paper, we address the solution to the execution of online transactions regardless of their failure due to disconnections.

2. PROTOCOL DESCRIPTION

2.1. Reliable Timeout Based Commit Protocol

This execution model has the Mobile Host (MH) and the Base Station (BS) communicating with each other through messages. The Mobile Transactions (MT) are initiated by the MH & are executed either at MH or at BS (BS is considered as coordinator-CO), Hence it uses distributed mode of execution between a MH & the data base servers-Fixed Cohort Unit (FCU) on the wired network at BS.

The designed algorithm [7] I.e. Transaction execution at Mobile Host (MH), initiates transaction at MH & fragmented it into set of sub transactions. A fragment of the transaction e_{i0} is executed at MH & the other fragments of transaction i.e. T_i- e_{i0} is at CO. While executing e_{i0} at MH it calculates the E_t & along with the Et of ei0 it sends T_i - e_{i0} to the CO for the execution of remaining fragments. On receiving (T_i-e_{i0}) the CO distributes set of fragments among various fixed cohorts (FCU) at the wired network. At the cohort's side, once after receiving the fragments they calculate the E_t's as per their requirements and sends them to CO. Once after receiving all E_t's, the CO calculates maximum Time (T_m= Max (E_{t0}, E_{t1},…..,E_{tn}) required to execute the transaction at MH & Participant FH(MH & FCU). A small value of E_t may generate a large number of extension requests.

The MH begins processing of e_{i0} & after successful completion, updates the local log. During processing, if any of the nodes needs to extend the E_t, similar to [4] they extend & sends the same copy to CO, then CO calculates T_m based on extended E_t's; if CO does not receive Commit message or if it receives an Abort message before Tm expires, then CO decides to abort T_i, else it decides to commit & send the same message to all, based on which the MH & FCU's updates their databases.

The second algorithm [7], i.e. Transaction execution at CO (at BS), the whole transaction T_i is executed at CO by originating at MH. The CO in turn sends fragmented transactions T_i (e_{i0}, e_{i1}, e_{i2},……, e_{in}) to all the participating FCU's (Cohorts) at the wired network. The part-FH calculates the E_t (Execution Time – E_{t0},E_{t1},E_{t2}…..,E_{tn}) of their respective fragments & sends them to the CO, then the CO calculates Maximum Time (T_m= Max (E_{t0}, E_{t1},…..,E_{tn}) Each FH begins processing of their respective e_i's & after successful completion, updates the local log. During processing, if it needs to extend E_t, FCU's extend & sends the same copy to CO. If CO does not receive Commit message or if it receives an Abort before T_m expires, then CO decides to abort T_i else decides to commit & send the same to all, based on which the MH & FCU's update their databases.

2.2. Handling Disconnections

The main characteristic of mobile computing is to support disconnections as a normal state of the system. With RTBC Protocol it is possible to handle transaction processing efficiently in case of disconnection. There are two kinds of disconnection, one is due to user request and the other is due to consequence of some mobile environment variations e.g. MH is out of the area covered by the wireless network or MH battery is exhausted. RTBCP tolerates above type of disconnections. The MH & all FCU's maintains their databases locally, hence they can continue computing with disconnection. The decision to the coordinator is sent when reconnection occurs. In the same way, the timeout assigned to each component transaction allows the MH to disconnect provided that the *vote* is sent before the timeout expires.

Generally, disconnection durations are frequent (0.001%) but of little duration so as a period of disconnection is often (in msec) less than the extended transactions deadline [1]. Therefore, when the connection is reestablished, the FH either sends or become ready to receive commit messages before the deadline expires from the participating members then it decides to commit else it aborts. The main objective is then to insure the availability and consistency of information in the database, in spite of disconnection.

The below algorithm designates the processing of transaction in case of disconnection with disconnection probability showing 1 among 1000(0.001 %) with Maximum disconnection period (MaxDT-Maximum Disconnection Time in msec) before new connection is established, Where disconnection period is lesser than the connection period.

///Algorithm for Disconnection with the 0.001% probability handling

1. While processing at MH & FH {
2. If ("disconnected before the start of processing but once after transmitting the set of fragments to CO")

   {Locally process & update the log and reconcile the results subsequently for the period of disconnection (MaxDT).}
3. Else If ("Disconnection has occurred before sending the results to CO")

   {Call Undo operation; Transaction executions are to be Undone}
4. Else If ("Disconnection has occurred after sending the results")

   {Call Redo operation; transactions execution are to be Redone}
5. Else {Commit all the processes & update into the log.}

   }
6. End

2.3. Failure Recovery

With the above algorithm it is possible to handle transaction processing efficiently in case of disconnection. The recovery of mobile transaction [5] which is executed at MH can be performed without any complications by maintaining logs locally at MH. Rolling back in case of communicated actions, the BS has to explicitly identify the actions need to be rolled back. The rollback algorithm [8] for mobile transaction environment uses the concept of immediate update recovery technique. The transaction manager creates Transaction table (Commit table) & Active table according to the log creation. In case of recovery to rollback the database, the recovery manager reads the records based on the type of log record, accordingly the active table transactions are undone & commit table transactions are redone.

The recovery model uses the logged data which is maintained using flash memory at MH and makes it available at any time during the transaction failure. Log records on the flash memory are organized depending on the checkpoints to provide fast references in the reconstruction of failed transactions.

3. PERFORMANCE EVALUATION

3.1. Simulation Model

To evaluate the performance of Reliable timeout based commit protocol, we developed a detailed simulation model. The components of our simulation model are shown in Figure 1. The simulator is implemented in J2EE; it consists of Mobile Host, which has a Mobile Transaction Generator (MTG) which generates transactions randomly to access the data items of the sample database, Transaction manager, Message Server (MS), Disconnection Predictor (DP), Data Manager and MEU to execute mobile transactions.

Fixed Host has Transaction Manager(TM), Message Server (MS), Data Manager (DM), and Execution Unit (FEU) for each transaction it coordinates and Fixed Cohort Unit (FCU) for each transaction that has submitted an operation to it.

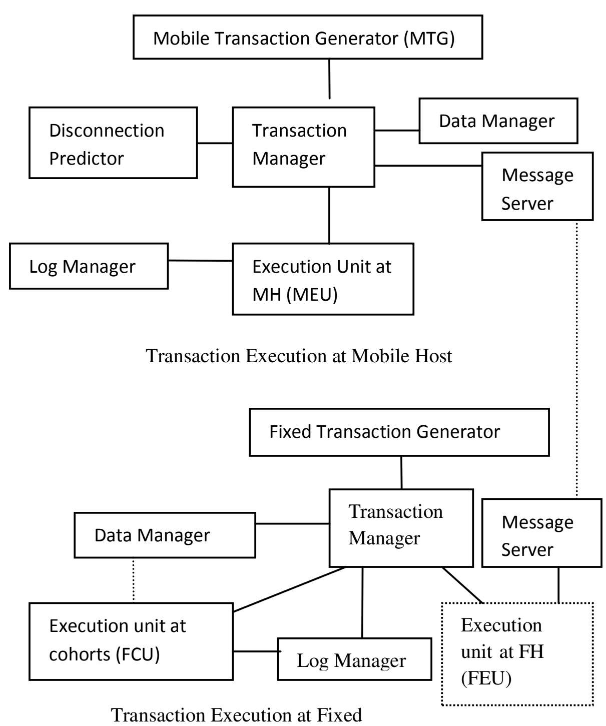

Figure 1. Components of simulation model

Transactions are generated by the MTG at MH, at each simulation run; at least 5 transactions are generated according to the parameters of the transaction model. In order to simulate the features of mobile computing such as frequent disconnection, and long-lived transactions, different time delays are introduced by MTG in generating transaction operations. They are submitted to the execution model for further processing. Deadline of each transaction is also performed by the TG.

The transactions generated at MTG are fragmented into set of sub transactions & part of each transactions are processed by MEU ,the Transaction Manager initiates a Mobile Execution Unit (MEU) at that host and sends a message to the relevant FH for the initiation of FEU at the coordinator site. The transaction manager at a FH that receives transaction initiation message first checks & sends the list of transaction fragments to be executed at FCU & initiates an FEU for the transaction execution of their respective fragments. MS is responsible for the transmission of messages between hosts.

Disconnection and reconnection of MH are initiated by the disconnection predictor. The execution of transaction operations is controlled by Execution unit at MH & Execution Unit at FH provides the execution of read and write operations via FCUs. Data manager allow us to access the data and it makes an updates to the databases on any write events. A summary of the parameters used in the simulation model is presented in above Table 1.

Table 1: Simulation Parameters

NFHosts	Fixed hosts	02
NMhosts	Mobile host	01
DBSize	No. of tuples in database	30 Tuples
Read /Write Ratio	No. of read & Write operations	70:30
UpTrProb.	Update transaction probability	0.25 - 0.75
WriteProb.	Write operation probability for update transactions	0.25 - 0.75
TrgnTm	Time in seconds to generate the transactions.	0 sec
ConnectInt	MH stays connected/ disconnected on the average interval time	5sec
DissconProb	Probability of getting disconnected MH	0.001msec
MCPUTime	Avg CPU time to process a fragment at a MH	50 msec
FCPUTime	Avg CPU time to process a fragment at a FH	50 msec
MsgCPUTime	Avg CPU time to process a message at a FH	2msec
NumfragMT	Num of fragments/mobile transactions	03

3.2. Results & Discussions

The disconnection may occur due to unavailability of wireless links, in such case the MH can no longer communicate with coordinator to perform any type of operation , this may impact the performance of the system. With related to our work, we have tried to use local logs to keep continuity of the processing in case of disconnections, so the performance of the system may not degrades.

In our experiment we have used the disconnection probability ranging from 0.001 to 0.005; it shows the probability of disconnecting a MH during the connected time period. Figure 2 shows the performance of the system in terms of Abort rate (due to disconnection & due to timeout during execution) & commit rate v/s Disconnection probability. As estimated, due to increase in disconnection probability the abort rate also increases, hence to some extent it may affect to the commit rate.

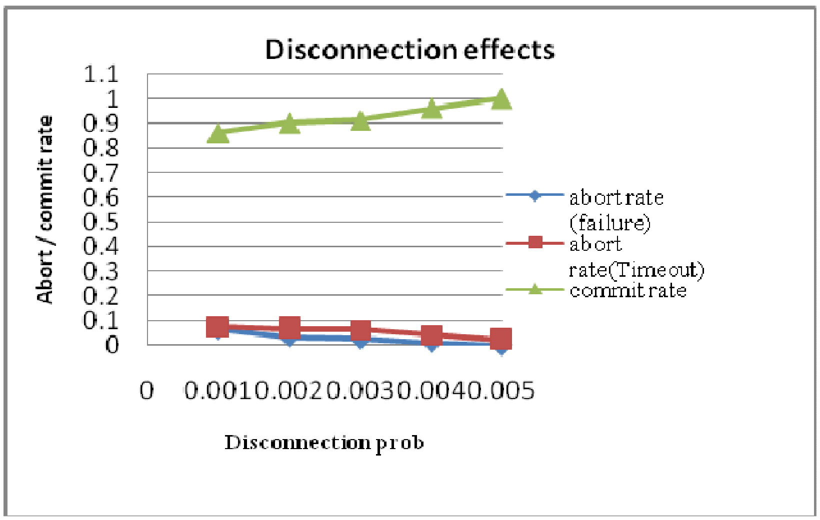

Figure 2: Disconnection probability V/S Abort & Commit Rate.

4. CONCLUSION

This paper proposes model for an efficient execution of mobile transactions where they deal with large set of distributed database with the disconnection probability of (0.001%). By making use of logged area at MH & FH we can process transactions locally during network disconnections which lead to the consistency & continuity of computing. As the applications are real time in nature, our simulation model aims to solve the problem towards managing frequent network disconnections. Hence this will be a new challenge in mobile computing where disconnections between user devices and the main sites occur frequently.

REFERENCES

[1] C. Bobineau, P. Pucheral, and M. Abdallah." A Unilateral Commit Protocol for Mobile and Disconnected Computing." In *PDCS*,USA, 2000.

[2] Z. Ding, X. Meng, and S. Wang ",O2PC-MT: A Novel Optimistic Two-Phase Commit Protocol " H.C. Mayr et al. (Eds.): DEXA2001, LNCS 2113, 2001.c Springer-Verlag Berlin Heidelberg 2001

[3] Christophe Bobineau, Cyril Labb'e, Claudia Roncancio, Patricia Serrano- Alvarado 'Comparing transaction commit protocols for mobile environments', in Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04) 1529-4188/04 IEEE

[4] V. Kumar, N. Prabhu, M. H. Dunham, and A. Y. Seydim. "TCOT- A Timeout-Based Mobile Transaction Commitment Protocol." *IEEE Transactions on Computers*, 51(10), 2002.

[5] patrica Serrano ,Claudia roncancico,Michel adiba 'A Survey of mobile transactions 'Distributed and parallel databases 16,193- 230,2004.

[6] M. M. Goreyand R. K. Ghosh "Recovery of Mobile Transactions" Proceedings of the 11thInternational Workshop on Database and Expert Systems Applications (DEXA'00)0-7695-0680-1/00 c 2000 IEEE.

[7]Bharati Harsoor, Dr.S.Ramachandram, 'Reliable Execution of Mobile Transactions' "This is my paper", proceedings of 2nd International conference on Wireless Information Networks & Business Information Systems (WINBIS- 10) .

[8] Bharati Harsoor, Dr.S.Ramachandram , 'Reliable Timeout based commit protocol' "This is my paper", proceedings of Third International conference on Network Security & Applications (CNSA-2010).

Authors

Mrs. Bharati Harsoor received her bachelor's degree in CSE (1995), Masters in Computer Science (2001). She is a Research Scholar at Osmania University, Hyderabad..She is presently working as Asst. Professor, Department of Information Science, PDA College of Engineering, Gulbarga, Karnataka State, India and Published papers in various National/International Conference and Journals. Her areas of interests include Mobile Computing, Databases, and Software Engineering. She is member of Institute of Electronics and Telecommunication Engineers (IETE).

Dr. S. Ramachandram (1959) received his bachelor's degree in Electronics and Communication (1983), Masters in Computer Science (1985) and a Ph.D. in Computer Science (2005). He is presently working as a Professor and Head, Department of Computer Science, University College of Engineering, Osmania University, Hyderabad, India. His research areas include Mobile Computing, Grid Computing, Server Virtualization and Software Engineering. He has authored several books on Software Engineering, handled several national & international projects and published several research papers at international and national level. He also held several positions in the university as a Chairman Board of Studies, Nodal officer for World Bank Projects and chair of Tutorials Committee. He is a member of Institute of Electrical and Electronic Engineers (IEEE), Computer Society of India (CSI) and Institute of Electronics and Telecommunication Engineers (IETE).