CONCURRENCY CONTROL FOR MOFLEX TRANSACTION MODEL

ALQEREM, AHMAD
Zarka Private University
Computer science department
Ahmad_qerm@zpu.edu.jo

DR. HUSSAK ,WALTER
Loughbourogh University UK
Computer Science Department
w.hussak@lboro.ac.uk

ABSTRACT

Tremendous advances in wireless networks and portable computing devices have led to development of mobile computing. Several mobile transaction models have been proposed to adhere with the nature of the mobile computing environment,Clusterd model[2],kangaroo model[3],pro-motion model[4] ,non of them fully supports the mobile computing requirements excepts the Moflex transaction model [1],but it lacks the concurrency control schema to ensure the correct execution of multiple Moflex transactions and avoidance of unnecessary aborts or compensation. In this paper we propose a scheduling protocol by exploit the concepts of F-serilaizability [2] to be the correctness criteria that maintain consistency of a multidatabase system accessed by concurrent Moflex transactions.

KEYWORDS

Mobile Multidatabase System, transaction management, concurrency control, Moflex transaction, MF-serializability

1-INTRODUCTION

Mobile computing is the ability of users to access the multidatabase system by mean of a wireless device. This type of computing is especially attractive to organizations that encompass a large geographic location. The mobile multidatabase is the integration of pre existing local databases that is accessed by one or more mobile devices. There are, however, hardware and software concerns that need to be addressed before any mobile system can be realized. In mobile computing the limited lifespan of the battery power of devices, limited wireless data bandwidth capabilities, and changing of the physical location of the mobiles devices that are accessing the data provide hardware obstacles that need to be addressed. Software concerns, which are directly related to aspects of the database, can involve data management, recovery, and transaction management. A lot of research have been done to design a transaction model that cope with the limitation of the mobile computing environment [2],[3],[4], [1], among those models ,the Moflex transaction model[1] includes not only the features for multidatabase system but also those for mobile computing environment. More over, Moflex transaction model allows the definition of location dependent query and the effective support of handover during the execution of subtransactions.

In this paper we propose an efficient concurrency control schema to mange the execution of multiple Moflex transaction, the paper is organized as follows: section 2 describe the mobile computing environment .section 3 involve the overview of the Moflex transaction model. Section 4 introduces the correctness criteria for Moflex transaction .section 5 gives the proposed concurrency control protocol .section 6 gives a brief conclusion of the paper.

2-MOBILE MULTIDATABASE ENVIRONMENT

It is important to identify and define the mobile computing environment. Based on that defined mobile environment, requirements as well as characteristics will be identified. Mobile computing environment includes: a wired network with fixed work-stations or fixed hosts (FH), mobile hosts (MH) and mobile support stations (MSS) [3] [4] [8] [9] as in figure 1.

Connection between MH and MSS is wireless network, this network is characteristic by it low bandwidth, error-prone and frequently disconnection. MSS and FH communicate with each other via reliable high-speed connection networks, which can be wired network or wireless network (within limited range, such as inside a building). The MSS is motionless. Mobile hosts can include broad types of mobile devices, typically laptop computers with high-speed modems. Works can be sharing between MH and FH. The role of MSS is not processing element but it is acting as an interface to help MH getting contact with relevant FH.

Each MSS responds for an area (called a cell) in which it will support all MH operate in this area. One MH can only connect to one MSS at any given time. A mobile host is moving from one area to another area when computation work is in processing, and sometimes MH requests to connect to a database or computing resource resided from a FH on fixed network. This work will be done with the help of MSS. Mobile support station will receive requests from MH, forward the requests to the responsible FH and return the answer from the FH to the MH. When a MH is leaving a cell controlled by a MSS, this MSS will perform a hand-off operation to transmit or forward all information related to this MH to next MSS. The next MSS in new cell will be ready to support the MH.

The general mobile multidatabase system is a collection of autonomous database connected to the fixed network. the respective database management systems continue complete control over their data. each database stores in an independent site of the fixed network. these database are in different environment and may use different data models, data manipulation languages, transaction management and concurrency control mechanism ,and so forth .thus a MMDBS can be viewed as a multidatabase system that supports mobile users.

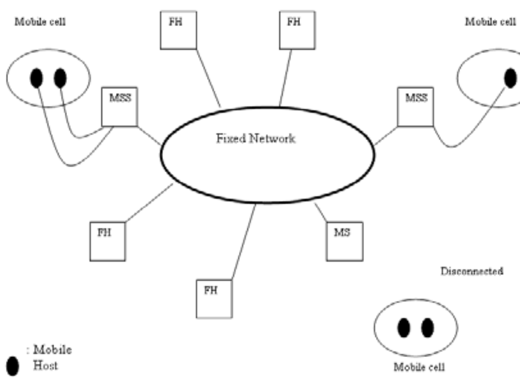

Figure 1. Mobile Computing Environment

3-MOFLEX TRANSACTION MODEL FOR MMDBS

A Moflex Transaction for mobile heterogeneous multidatabase systems is formally defined as follows:

Definition 3.1 A Moflex Transaction T is a 7-tuple (M, S, F, $\prod$, H, J, G)

M : $\{t_1, t_2, \ldots, t_n\}$, the set of all subtransactions of T where t_i is either compensatable(C) or noncompensatable(NC).
S : the set of success-dependencies in M
F : the set of failure-dependencies in M
$\prod$: the set of external-dependencies(P, Q, L) on M
H : the set of hand-over control rules on M
J : the set of acceptable join rules on M
G : the set of all acceptable goal states of T

A Moflex transaction has a built-in set of dependencies, acceptable goals and rules.

Three kinds of dependencies are considered:
- ***Success:*** A sub-transaction is executed only if a related sub-transaction successfully completes.
- ***Failure:*** A sub-transaction is executed only if a related sub-transaction fails.
- ***External conditions:*** A sub-transaction is executed only if an external condition is verified. An external condition can concern time, cost, or location.

The set of acceptable goals is the set of the final sub-transaction execution states that the user considers proper for his application.

Hand-Over Control rules determine the sub-transaction execution policy when the mobile unit
changes from a cell to another. Depending on whether or not the sub-transaction is compensatable and location-dependent, four different execution policies can take place :
- ***SplitResume*** (compensatable and location-independent). The sub-transaction is split.
The already executed portion of the sub-transaction is committed. The remaining execution is continued as a new sub-transaction on the new base station.
- ***Restart*** (location-dependent, either compensatable or not). This is the default operation.The sub-transaction is restarted on the new base station.
- ***SplitRestart*** (location-dependent and compensatable). The executed portion of the subtransaction is committed on the old base station and the sub-transaction is restarted as a whole on the new base station.
- ***Continue*** (location-independent and non-compensatable). The execution continues on the new base station without interruption.

The system model employed for the Moflex Transaction Model is depicted in Figure 2. the system is built on heterogeneous, autonomous multidatabase systems.

The mobile heterogeneous multidatabase system consists of three-layers: Mobile Host(MH) layer, Mobile Support Station(MSS) layer and heterogeneous multidatabase system(HMDBS) layer. In MH layer, the user on MH defines Moflex Transactions and submits them to Mobile Transaction Manager(MTM) of current wireless cell in the MSS layer. MTM coordinates the execution of the submitted Moflex Transaction.

HMDBS is a logically integrated system to provide information to not only users on fixed networks but also users on wireless networks. In Figure 2, HMDBS is identical to the Flexible Transaction processing environment[9] In HMDBS, Flexible Transaction is submitted to the Global Transaction Manager(GTM) for the coordination of the executions at several local database systems(LDBSs) such like global concurrency control, global commitment control, and replica control. Local Execution Monitor(LEM) is the communication interface between MTMs and LDBSs.

Definition 3.2 If Moflex Transaction T consists of n-subtransactions $\{t_1, t_2, \ldots, t_n\}$, the execution state x of T is an n-tuple $(x_1, x_2, \ldots, x_n)$, where xi = 'N': not submitted for execution;

'E': currently being executed;

'S': successfully completed;

'F': failed or completed without achieving its objective.

If Moflex Transaction T consists of n-subtransactions $\{t_1, t_2, \ldots, t_n\}$, the execution state x of T is represented as $(x_1, x_2, \ldots, x_n)$. 'N' means that subtransaction t_i has not been submitted for execution. 'E' stands for that subtransaction t_i is currently being executed. 'S' represents that subtransaction t_i has successfully completed. If subtransaction t_i is compensatable, x_i = 'S' means that t_i has been committed. Otherwise, i.e. if t_i is non-compensatable, x_i = 'S' means that t_i is in prepared-to-commit state of two-phase commitment protocol(2PC). 'F' means that subtransaction t_i has failed or completed without achieving its objective. If subtransaction t_i is compensatable, x_i = 'F' means that t_i has been aborted or compensated. Otherwise,
x_i = 'F' means that t_i has been already aborted.

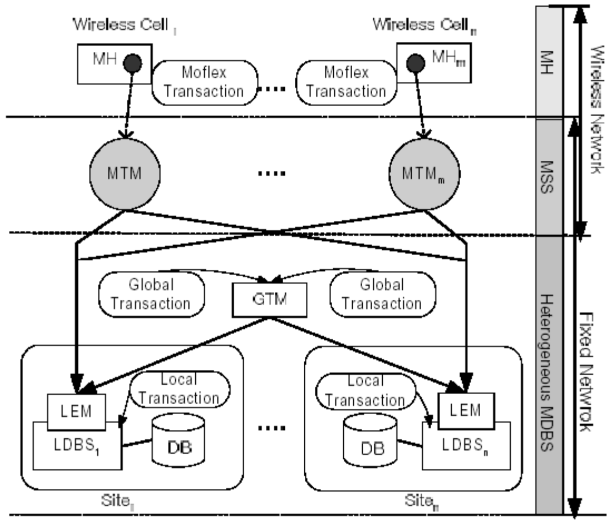

Figure 2. The Architecture of Mobile Heterogeneous Multidatabase system

The MTM in MSS layer controls the global execution of Moflex Transaction T by the algorithms that have been discussed in [1].

EXAMPLE OF MOFLEX TRANSACTION

As an example, an Emergency Patient Transmission Service in a domain is depicted in Figure 3.

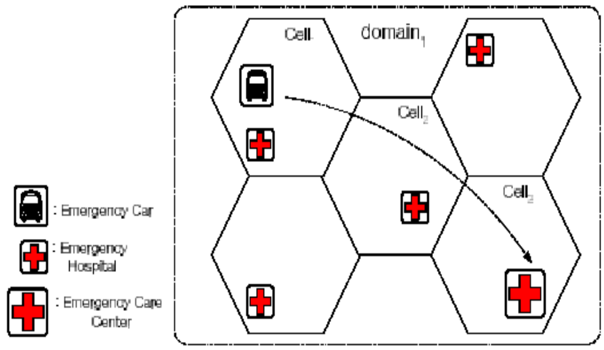

Figure 3. An Emergency Patient Transmission service

Let us assume as follows. In the system, an MH is the ambulance vehicle equipped with mobile computers and the MH has to quickly find the proper emergency hospital for a patient. If it fails to find a proper emergency hospital in cell1, it tries to find a proper emergency hospital in cell2. If it succeeds to find a proper emergency hospital in cell1 or cell2, the geographical information for the hospital is provided to the MH. If it fails to find the proper emergency hospital in cell1 and cell2, the patient is transmitted to the default Emergency Care Center in cell3. The MH sends the current emergency status of the patient to the selected hospital in order to prepare the appropriate medical service for the patient. And, the MH retrieves the patient record at the hospital insurance system.

The activities of Emergency Patient Transmission Service are modeled by a Moflex Transaction as follows:

Example: The Moflex Transaction for emergency patient transmission service

t_1 : find the proper hospital

t_2 : transmit to the default emergency care center

t_3 : send the current emergency status for the proper service

t_4 : get the geographical information for the hospital

t_5 : get the patient record

The formal definition of the Moflex Transaction for the above example is depicted as follows

The alternative to subtransaction t_1 is t_2 when t_1 fails.

M = {t1(C), t2(C), t3 (NC), t4(C), t5 (C)}

S = {t1 $\prec_s$ t_3, t_2 $\prec_s$ t_3, t_1 $\prec_s$ t_4}

F = {t1 $\prec_f$ t_2}

$\prod$ = {L}

L = {t1, t_4}

H = {restart (t_1), continue (t_2), continue (t_3), split resume (t_4), continue (t_5)}

J = {user (t_4)}

G = { (S , - , S , S , S),(- , S , S , - , S) }

In the representation of G, 'S' stands for the successful execution of the corresponding subtransaction and '-' means that the execution state of the corresponding subtransaction does not affect the decision, whether the current execution state is equivalent to one of the acceptable goal states.

From the above example, we can get the point that the proposed Moflex Transaction Model is able to provide the flexibility on the definition of mobile transactions. In addition, the proposed Moflex Transaction Model [1] is able to effectively support the mobility of MH in the execution of mobile transactions.

4- SERIALIZBILITY WITH MOFLEX TRANSACTIONS

The MTM is responsible for coordinating the execution of the Moflex transaction so

That each MTM maintain an MSEG graph that ensure the correct execution of concurrent Moflex transactions on the MSS that host the MTM .when the hand over occur the communication between different MTM are needed to update the MSEG graph based on the edges and node insertion and deletion rule. this well be shown in the following algorithm:

Definition 4.1 Moflex Subtransaction Execution Graph. The MSEG of a set of Moflex transactions in schedule S at MTM is a directed graph whose nodes are Moflex subtransaction and compensating subtransaction for those Moflex transactions and whose edges $t_i \rightarrow t_j$ indicate that t_j must serialize before t_i due to the success ,failure or conflict .

Let AC(t) denote the set of data items that t accesses and commits, RC(t) denote the set of data items that t reads and commits, and WC(t) denote the set of data items that t writes and commits. compensation-interference free property on schedule S is define in [2] as follows:

Definition 4.2 (**Compensation-interference free**)

A global schedule S is compensation-interference free if, for any subtransaction t_j which is serialized between a subtransaction t_i and its compensating transaction ct_i in S,

$WC(t_i) \cap AC(t_j) = Ø$.

Based on the above propriety the new correctness criterion called MF-serializability was defined as follows:

Definition 4.3 (MF-**serializability**)

Let S be a schedule of a set of Moflex transactions . S is MF-serializable if it is conflict serializable and compensation-interference free.

5-CONCURRENCY CONTROL FOR MOFLEX TRANSACTIONS

The Moflex Transaction Model [1] generalizes the Flex Transaction Model to support

location-dependent sub-transactions in heterogeneous multi-database environments. It supports

hand-offs between base stations when the mobile unit moves from one cell to another.

The mobile unit creates a Moflex transaction and submits it to the *Mobile Transaction Manager*

(MTM) residing on the corresponding base station. The MTM sends the steps of the Moflex

transaction to the respective local database systems and coordinates their global commitment. In each database system, a *Local Execution Manager* acts as an interface with MTMs in order to detect global conflicts.

Node Insertion Rule:

Insert a node for each subtransaction defined for t_i For each compensatable subtransaction insert a node ct_i.

Edge Insertion Rule:

For subtransaction t_i, where edge insertion does not cause a cycle:

1. For each previously-scheduled t_h, $h < i$,.
if $WC(t_h) \cap AC(t_i) \neq \emptyset$, insert edge $t_i \rightarrow t_h$
2. For each previously-scheduled ct_h , $h < i$,
if $WC(t_h) \cap AC(t_i) \neq \emptyset$, insert edge $t_i \rightarrow ct_h$.
3. If t_i is compensatable, insert edge $ct_i \rightarrow t_i$.
4. If $t_i \prec_s t_j$ then insert edge $t_i \rightarrow t_j$ and label it as S .
5. If $t_i \prec_f t_i$ then insert edge $t_i \rightarrow t_j$ and label it as F

The first two edge insertion cases ensure MF-serializability The third rule ensures that the invalid subtransaction precedes its compensating transaction. The rest of the cases ensure that, execution of all subtransactions must ensure success and failure dependencies as defined in the Moflex transactions in order to reach one the acceptable goal states ,

Nodes and edges are deleted from the MSEG according to the following rules:

Node-Edge Deletion rule:

Case 1,2 and 3 are chosen based upon the acceptable goal states are reached or the subtransaction that involve in the set of success and failure dependency are commit or abort or the handover occur.

1-If the Moflex transactions reaches one of its acceptable goal states then delete all the nodes and edges representing this transaction.

2- Check the subtransaction state

Case S:

Delete the node representing this subtransaction and all the edges labeled F related to this node and their nodes.

Case F:

Delete the node representing this subtransaction and all the edges labeled S related to this node and their nodes.

3- If the handover occur check the handover rule then Switch (handover rule):

Case split resume or split restart:

Decompose the node into tow adjacent nodes n_{j1} and n_{j2} and send n_{j2} to the new MTM and wait for response (commit or abort)

Case restart:

Delete the node from the MSEG graph in the current MTM and send it to the new MTM.

And wait for commit or abort.

Case continue

Do no action just wait for response (commit or abort) then apply the usual deletion rule

The operations of a Moflex subtransaction of T_i are submitted to the LEM according to the following rule.

Operation Submission Rule:

Submit operations of a subtransaction (including begin and commit) to its LEM only if its node in the MSEG has no outgoing edges and the external dependency predicates are satisfied.

Lemma 1 The MSEG protocol for scheduling Moflex transaction at all MTM maintains Mf-serializability and avoids the cascading aborts.

Proof

Assume that we have MTMS={MTM_1 , MTM_2 ,…….., MTM_n } each of them contain an $MSEG_1$, $MSEG_2$,……….. , MSEGn graphs ,then by the third deletion rule all of these MSEGs graph have no common node or edges so the union of all these graph will give us the MSEG graph that satisfy the MF-serializability conditions for all Moflex transaction executed at different base station even if the hand over occur or more than one MTM involve in the execution of the same Moflex transaction also by ensuring the compensation interference free propriety in the edge insertion rule we can avoid the cascading aborts.

6-CONCLUSION

In this paper we based on the notion of F-serializability[2] to develop a new correctness criteria for Moflex transaction model called MF-serializability with a scheduling protocol that maintain the correct and efficient execution of Moflex transactions called MSEG.

The MSEG protocol for scheduling the Moflex transaction maintain MF-serializability

by ensuring the compensation interference free propriety in the edge insertion rule

that avoid us the cascading aborts, on the other hand the edge deletion rule in case of F and S increase the performance by allowing the conflicting subtransaction from different Moflex transaction to be ready for execution by removing the edges label F or S depend on the status of the subtransaction which complete their execution either by success of failed. So The MSEG protocol for scheduling the Moflex transaction maintain MF-serializability and avoids the cascading aborts and increase the performance.

REFERENCES

[1] Kyong-I Ku; Yoo-Sung Kim *Moflex transaction model for mobile heterogeneous multidatabase systems* Research Issues in Data Engineering, 2000. RIDE 2000. Proceedings. Page(s): 39 –45

[2] Aidong Zhang , NY 14260 Marian Nodine , RI 02912 Bharat Bhargava , IN 47907, Global Scheduling for Flexible Transactions in Heterogeneous Distributed Database Systems,IEEE,2001.

[3] V.K. Murthy *Seamless Mobile Transaction Processing: Models, Protocols andSoftware Tools*. In Proc. 11th ACM Annual Symposium on Applied Computing, pages 171--181, 1997.

[4] Madria, S.K., Bhargava, B *A Transaction Model for Mobile Computing* Database ,Engineering and Applications Symposium, 1998. Proceedings. IDEAS'98.

[5] Margaret H. Dunham, Abdelsalam Helal, Santosh Balakrishnan *A Mobile Transaction Model that Captures Both the Data and Movement behavior*

[6] A. Elmagarmid, Y. Leu, and M. Rusinkiewicz, "A Multidatabase Transaction Model for INTERBASE," In Proceedings of the International Conference on VLDB, Aug. 1990.