Location dependent transaction for mobile environment

This item was submitted to Loughborough University's Institutional Repository by the author.

Citation: ALQEREM, W. and HUSSAK, W., 2006. Location dependent transaction for mobile environment. IN: Proceedings, IEEE International Conference on Information and Communication Technologies, ICTTA '06. Volume: 2, 24-28 April 2006, pp. 2926 - 2928

Additional Information:

- This is a conference paper [© IEEE]. It is also available from: http://ieeexplore.ieee.org. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Metadata Record: https://dspace.lboro.ac.uk/2134/4166

Version: Published

Publisher: © IEEE

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
Abstract
With recent advances of mobile and portable devices, more than one billion cellular phones in the world joined by other wireless handheld computing devices like personal digital assistants (PDAs) or pocket PCs, with this number of users there are significant opportunities for mobile commerce growth. Although mobile commerce enables access to goods and service regardless of the location of either buyer or seller, in many situations the specific location of the buyer and seller is critical to the transaction [1]. Also the time for transaction execution become increasingly important not from performance point view but also from the corresponding relationship between the data and location especially when the mobile user change its location dynamically.
In this paper we aim to introduce a mobile transaction model that takes into consideration the location dependent transaction and the time constraint for mobile transaction execution.

1. Introduction
In this paper we proposed a framework for a location dependent mobile transaction that allow the mobile transaction to access a multi database and change its location during their execution and this framework is compatible with UMTS all IP Network and use the same relocation schema [2]. The system architecture of our model is shown in the figure 1, each level contain the data abstraction (Meta data) of the below level. The paper is organized as follow. section 2 introduce the formal model for location dependent mobile transaction in section 3 we give some clarifying example in section 3 we introduce the concurrency control t schema for managing multiple location dependent transaction , section 4 conclude our work.

2. Formal model
Location dependent mobile transaction is defined by a 3-tuple (S, D, P) where S is a set of sub- transaction and D represent the set of dependencies that define over a power set of S and P is a set of predicate (T,L,Q) associated with each subtransaction , where T is the deadline for location dependent subtransaction execution and this will not be exceeded and L which represent the type of each subtransaction if its location dependent or not. Q which represent any other condition to be satisfied before transaction execution , an optimistic concurrency control technique based on the timestamp ordering are used to deal with different concurrent subtransactions from different location dependent mobile transaction that investigate the set of predicate P, in the scheduling protocol, a simulation study have been conducted and show how this model effectively deal with a mobile transaction which access multibase system while moving from one location to another.

Figure 1
The set of dependency D can represent any sort of transaction model starting from flat transaction which constitute of a set of read write operation with a partially ordered relationship (i.e. there is one subtransaction and D is empty ) to the advanced transactions model depend on the semantic of the application which used for Q may used to help increasing concurrency (i.e. there are two subtransaction from different mobile transaction tij, tkj and tij need a data items S and tkj need a data items U and S is contained in U, or the region where S is a
valid data is cover the region where the U is valid, or the
time interval for the tij is contained in the time interval
needed to accomplish (tkl)

Each location govern by a manager which is responsible
with other locations managers to communicate with the
GTM of the database to execute and commit the
transactions who cross over more than one location

Exmaple:
The user X is looking for booking a restaurant at 2:00
pm at lunch time with budget 10 $ and after that he want
to go to cinema to see a thrill movie at 4:00 pm also he
have a condition that the restaurant and the cinema must
have a park and the distance between the restaurant and
the cinema not exceed 2 km and this will start at the
time of issuing the transaction then he specify that the
confirmation of booking these facilities must be within 5
km of his current location and not exceed 1:00 pm.

T1: booking restaurant
T2: booking cinema
T3: check park availability
T4: check credit card availability

S = {t1, t2, t3, t4}
D = {t4-t3-t1, t4, t3, t2}, {RL - Cl < 2 km} where RL,
CL are a restaurant location and cinema location
respectively
P1 = (2:00 pm, 5km, 10$)
P2 = (4:00 pm, 5km, thrill movie)

Now depend on the user current location and the meta
data with their location and time condition specifies by
the transaction ,these data have to send to the most
generic node that meet all the requirement for this
transaction starting from the closet node to the user
current location and then further to the upper generic
node

The concurrency control algorithm based on time stamp
ordering are introduce to solve the scheduling problem
arise from this complicated situation[3] ,we assume here
that the global data base represent more than one agency
for each special type of facilities (i.e. the restaurant
agency contain all information about all the restaurant in
the location covered by this agency)

So the problem arise when there is more than one
location dependent transaction looking for the same
facilities and the location specified by their transactions
are overlapped in term of time and region, so the
algorithm must be solve the conflict based on criteria
specified in the predicate for each subtransaction, in
addition to that we need to investigate the intra
dependencies between the subtransaction that constitute
the transaction to increase the concurrency and ensure
the global serializability for all transactions

3. Composite Timestamp Ordering
Algorithm

The Composite Timestamp Ordering CTO imposes an a
priori total order on all multi location transactions [3]
The CTO algorithm assigns and validates timestamp
orders as follows all delegated subtransactions (the
subtransaction that send to database component under
the same multi database agency) are tagged as multi
location and inherit the global timestamp from their
parent transaction

Untagged transactions that enter a location
multidatabase are new transactions submitted directly
from an application New transactions are first
decomposed If any of the resulting subtransactions are
delegated to other multidatabase location then both they
and their parent are tagged as multi locations and
assigned the same unique timestamp Tagged transactions
are delegated from other multidatabase location These
transactions are first validated by checking that their
timestamp is not older than that of the last committed
transaction Tlast If it is older it is aborted _ Otherwise it
is decomposed All resulting subtransactions inherit the
tags and the unique timestamp from the parent and are
submitted to component databases for execution

The CTO algorithm ensures that new and preparing
delegated subtransactions execute and commit in
timestamp order The global concurrency control
scheduler of each location maintains a multi location
commit order list MLOL The MLOL is never empty
because it always maintains Tlast as a node Every multi
location transaction is added to the MLOL sorted on
their timestamp value and placed in a running state

When any subtransaction on the list is ready to prepare it
first checks if it is older than Tlast If it is older it is aborted
Otherwise it checks if any younger transactions are
prepared if so it is placed in a waiting state and
rechecks after a set interval or after a prepared
transaction is forced to abort through the 2PC protocol

Once a transaction passes the validation step it prepares
enters a prepare state and waits for a commit signal from
its 2PC coordinator On receiving a commit a prepared
transaction aborts all older transactions in the running
and waiting states then commits If it receives an abort
signal it is removed from MLOL and signals all
older transactions in the waiting state
4. Conclusion
The significant issues such as location transparency, scalability, performance, and administration make multidatabase location composition inevitable. We have shown that scheduling transactions that span multiple multidatabase location with predefined predicate associated with each subtransaction in term of time and location imply a matter of guaranteeing transaction atomicity and serializability at each multidatabase location. To guarantee global serializability in a composite multidatabase environment, ordering constraints must be imposed on multi location transactions. We presented a general transaction model that may fit for any type of application in the mobile environment and proposed a transaction scheduling algorithms based on the time stamp ordering that ensure the serializability of more than one location dependent transaction accessing the same common multi databases based on their location.

References

[2] Anna Ha’c, Senior Member, IEEE, and Bo Liu; Database and Location Management Schemes for Mobile Communications; *IEEE/ACM TRANSACTIONS ON NETWORKING*, VOL. 6, NO. 6, DECEMBER 1998