Mobile Transaction Models Framework

Andrés Coratella
Universidad de Buenos Aires, Argentina

Miguel Felder
Universidad de Buenos Aires, Argentina
Pragma Consultores

Roberto Hirsch
Universidad de Buenos Aires, Argentina

Eduardo Rodriguez
Universidad de Buenos Aires, Argentina

INTRODUCTION

Currently, mobile technology is undergoing a high growth stage, allowing for an increasing plethora of mobile devices (handheld PCs, handsets, etc.) and daily access to distributed resources and information.

This availability entails the requirement for transactional capabilities adapted to the specific characteristics of the mobile environment without losing the consistency and reliability guarantees of traditional online transactional processing (OLTP) systems.

One of the objectives of this work was the study of transactional models applied to mobile environments. After analyzing the state of the art, we observed that none of the models covers all the necessary characteristics for current requirements, and therefore, we propose a framework that allows us to capture and compare the main features to be taken into account in these models.

BACKGROUND

Transactional Systems' Requirements for Mobile Environments

Below we describe the requirements needed for a transactional system in order to be applied to a mobile environment (Tewari et al., 1995; Dunham et al., 1997; Lee et al., 1997):

- ***Ability to distribute the transaction's processing***: Due to memory, power processing, and battery limitations of the mobile hosts (MHs), it may be necessary to execute certain sections of the transaction in the mobile support station (MSS).
- ***Share the state and the partial results***: Because, as was stated in the previous item, parts of a transaction can be executed in the MH, while others run in the MSS, items must be shared.
- ***Capture the movement of mobile transactions***: Due to the physical movement of the MHs, it is necessary to transfer a transaction's control as it moves from cell to cell.
- ***Support long-lived transactions***: This is required, because some processes can take a considerable amount of time, and besides, the search for a computer that has physically moved from a cell can be a time-costly operation.
- ***Support long disconnection periods***: Recall that disconnections can be caused by physical problems, or simply by the MH's own decision. For an MH to continue operating despite being disconnected from the network, it may be necessary to maintain local copies of the data needed (*caching techniques*).
- ***Support partial failures and provide different recovery strategies***: These failures can be caused by battery problems, static electricity, accidental computer turnoffs, etc.

Mobile Transactional Models

The use of transactions in a mobile environment differs substantially from the use of transactions in centralized or distributed systems. The main differences are the high disconnection frequency and the mobility of the transactions. Therefore, transactional models and commit coordination protocols must be revised to take into account the mentioned differences (Dunham et al., 1997).

There are models that explicitly support mobility, such as Kangaroo Transactions (Dunham et al., 1997), which creates a subtransaction on every MSS that the user passes by, establishing a link between them so as to move the data as the user moves. It can use compensations, if necessary, in case of failures. Another example of this kind of model is Moflex (Ku et al., 1998), which allows for the definition of parameters that specify, in a flexible way, the behavior of the transactions. A transaction is made up of subtransactions that can be either compensatable or not. It is possible to define the behavior these transactions will follow, such as the success or failure dependencies they must maintain with other subtransactions, conditions regarding the geographic location, etc. When a transaction faces a handoff, it can behave as specified by the user through a series of rules. Other models, like reporting/cotransactions (Chrysanthis, 1993) allow explicit sharing of partial results between active transactions, while transactions migrate from MSS to MSS as the MH moves from cell to cell, in order to minimize communication costs. Finally, the prewrite model (Madria, 2001) incorporates a prewrite operation before a write operation and also supports mobility.

Another group of models supports disconnections. An example is isolation-only transactions (IOTs) (Lu et al., 1994, 1995), designed to allow disconnected operations in mobile computers, detecting read/write conflicts based in limitations of the serializability. When a transaction commits, results are published only if all the accessed data maintained a connection with the server; otherwise, results are only locally visible, and the transaction is validated at reconnection time. In the weak/strict transactional model (Pitoura et al., 1995), transactions can also execute in disconnected mode, but there are special versions of read and write operations (called *weak read* and *weak write*) that operate on local copies of data items, which must be checked and eventually published at reconnection time. Data clusters of computers are defined that enforce the locality and validity concepts. There is another model, called planned disconnection (Holliday et al., 2000), with the main innovation of planning the disconnections that the user will perform, so as to minimize disruption to the remaining sites. A *planned disconnection* is defined as a disconnection where the user informs the system of his or her intention to disconnect and reconnect in an orderly manner. Finally, it is important to note that the prewrite model (described in the previous paragraph) also supports disconnection.

For more information on these models, please read Coratella et al. (2003).

COMPARISON FRAMEWORK

The idea behind our comparison framework is to act as a guide that should allow us to capture and compare the main characteristics to be taken into account in mobile transaction models.

Definition

Here, we describe the main characteristics that must be taken into account in order to define a transactional model capable of being applied to a mobile environment, trying to take full advantage of its peculiarities.

Relating Physical Aspects

- ***Mobility support***: Maintain the transaction's execution, even though the computer moves from cell to cell.
- ***Disconnection support***: The ability to execute transactions even when the mobile computer is disconnected, using caching techniques.
- ***Replication support***: Support the replication of information to have a lower communication cost.

Relating Transaction Execution

- ***Place of execution***: Some models execute transactions at the MSS, while others execute them at the MHs.
- ***Compensable transactions***: Compensable transactions allow the system to partially commit changes, because they can be later reverted if the whole transaction aborts.
- ***Conditions of execution***: Conditions to be evaluated before, during, and after the execution of the transaction. These conditions can be based on time, etc.
- ***Efficient concurrency handling***: In mobile environments, there is a higher probability that a transaction will become a long-lived transaction (LLT), so the concurrency must be handled efficiently, in order not to lock an object for a long time (for example, Field Call).

Relating the Model's Adaptability

- ***Take advantage of the geographical localization of the mobile unit***: A transaction could specify location-dependent conditions in order to modify its behavior. Furthermore, a handoff may not be