Revising Transaction Concepts for Mobile Computing *

Position Paper

Evaggelia Pitoura

Department of Computer Sciences
Purdue University
West Lafayette, IN 47905
email: pitoura@cs.purdue.edu

Bharat Bhargava

Department of Computer Sciences
Purdue University
West Lafayette, IN 47905
email: bb@cs.purdue.edu

Abstract

It is expected that in the near future millions of users will have access to on-line distributed databases through mobile computers. The restrictions imposed by the nature of the wireless medium and the resulting mobility of data consumers and data producers make traditional transaction models inadequate. In this paper, we investigate means for providing transaction support appropriate for mobile environments. Specifically, we define a model that ensures a weaker notion of consistency, accounts for mobility, and provides for recovery.

1 Introduction

Recent technological advances have provided users with the possibility of accessing information through wireless connections. Today, when users move, unplug their computer from the local area network, transport it, and plug it back to the local area network at their destination. Wireless technology provides users with the ability to retain their network connection even while moving. The resulting distributed environment is subject to restrictions imposed by the nature of the wireless medium, and the mobility of users.

Distributed systems that support mobility consist of two type of entities, mobile and fixed hosts [8]. Some of the fixed hosts, called *base stations*, are augmented with a wireless interface to communicate with mobile hosts. The geographical area covered by a base station is called *cell*. Each mobile host can directly communicate with one base station, the one covering its current geographical area.

Research in networking, communication, and file systems has focused on providing the necessary infrastructure for realizing wireless systems. While this kind of infrastructure work is essential, there is also a need to study how data management applications for such systems should be designed and implemented [8, 7, 12]. Our research concentrates on providing transactional support appropriate for mobile computing.

In general, mobile users will desire access to private or corporate databases that will be stored at mobile as well as static hosts, and queried and updated over the wired and the wireless network. For example, insurance agents may interact through their mobile station with a database storing consumer records, while traveling salespersons may access inventory databases.

Transactions have been used to ensure consistent data management and atomic and isolated user interactions. While a model for transaction processing for mobile computing has not evolved yet, it is apparent that some notion of transaction is necessary for ensuring correctness. The users of a mobile host need to have an idea of the validity of the data they read, some assurance that their updates will be saved and an understanding of the interaction of their work with the work of others concurrently accessing the same data.

In traditional database systems, users interact with the database through transactions, which are atomic, consistent, isolated and durable. In the recent past, many of these properties have been reconsidered as they were found unsuitable for advanced applications such as CAD databases, cooperative environments, and control flow systems [3]. What are the appropriate properties for a transaction involving the wireless medium?

In this position paper, we present a brief outline of our ongoing research on the effect of mobility on transaction systems [13]. We define weaker notions of consistency and *weak transactions* that support disconnected operation, and partial connectivity, and are capable of modeling operations on imprecise data, such as location data. Then, we introduce *transaction relocation* for modeling mobility. Finally, we propose the notion of *transaction proxies* for supporting recovery.

The remainder of this paper is structured as follows. In Section 2, we study the structure and properties of mobile transactions. Section 3 introduces our notion of consistency. Transaction relocation is discussed in Section 4, while recovery is presented in Section 5. In Section 6, we compare our work with related research,

*in Proceeding of the IEEE Workshop on Mobile Systems and Applications, Santa Cruz, CA, Dec 94

and in Section 7, we conclude with a brief summary.

2 Modeling Transactions

A mobile transaction is a distributed transaction, where some parts of the computation are executed on mobile and some parts on nonmobile hosts. The use of the wireless medium and the resulting mobility of data consumers and producers affect transaction processing in various ways.

The employment of wireless connections results in transactions that are long-lived, because of long network delays. Moreover, mobile transactions involve sites that are only intermittently connected to the rest of the network. The reason is that users of mobile hosts often avoid the employment of the wireless medium since wireless connections are expensive both monetarily and in terms of power consumption, or because no networking capability is available at their current location. In addition, for some technologies, such as cellular modems, there is a high start-up charge for each communication. Cost-effective transaction management for such technologies may adopt the approach of supporting few long-lived session-based transactions instead of many short-lived transactions. Finally, mobile computation is more error-prone, because of frequent disconnections and because mobile hosts are more susceptible to accidents than fixed hosts.

Mobility results in transactions that access heterogeneous information systems. Furthermore, while in static designs the location of all users is fixed, in mobile environments it changes frequently. In consequence, mobile transactions access fast-changing (possibly imprecise) location data. In addition, any notion of locality changes with time. Finally, transactions may involve data that are being dynamically relocated.

In the following we discuss how these factors affect the structure and properties of transactions.

2.1 Transaction Structure

Traditionally, transactions are modeled as sequences of read and write operations, which have a *flat* structure with a single begin and a single abort or commit point. In a mobile environment, transactions model complex, long-lived activities than involve multiple and possibly heterogeneous databases.

Consider for instance a transaction that models the purchase of an item. A traveling salesman first checks the availability of the product, then the credit history of the prospective customer, and only if the result of both checks is satisfactory, he sells the item and credits the customer. This transaction involves accessing a number of different databases and requires the specification of a number of complex dependencies between the subtasks of the transaction. Moreover, during the execution of the transaction, the salesman may be temporarily disconnected from the network or move to a different physical location.

As a consequence, a more flexible transaction model is desirable [13]. We model transactions using an open-nested model [4]. According to this model, a transaction is viewed as composed of a number of subtransactions, which are related with a number of user-defined or application-specific dependencies. Each subtransaction is either a flat transaction or an open-nested transaction. The dependencies may for instance specify temporal conditions on the execution of a subtransaction or contingency transactions, e.g., transactions that are executed as alternatives when a specified transaction fails to commit.

2.2 Transaction Properties

In this section, we discuss briefly the properties of subtransactions in mobile environments. Traditionally, transactions are assumed to be atomic, consistent, isolated and durable (or in other words satisfy the ACID properties). Specifically, each transaction is *atomic* when either all or none of its operations are executed, *consistent* when its execution maintains database consistency, *isolated* when it does not observe partial results of any other transaction, and *durable* if its results become permanent after commitment.

Atomicity is a very restrictive property for mobile transactions, since they are error-prone and long-lived. In our open-nested model, this can be replaced by specifying atomicity requirements at the subtransaction level. The assumption of consistency is also very strict. Mobile transactions may read out-of-date local data while they are disconnected or imprecise location data when changes in location data are not constantly reported. We further discuss consistency in Section 3. The isolation property is application-specific since for some applications sharing of results between concurrently executing transactions may be desirable and is not directly affected by mobility. Finally, due to the fact that mobile transactions are error-prone durability may be too hard to enforce. We discuss that in Section 5.

Relocation is a new property that mobile transactions may support and is discussed in Section 4.

3 Data Consistency

A mobile host is only intermittently connected to the rest of the network. Thus, it is crucial to allow a mobile host to operate even while disconnected. Furthermore, since network bandwidth is a scarce and expensive resource in a mobile environment, transaction processing schemas should reflect a much greater concern for bandwidth consumption and constraints than schemas for nonmobile environments. Maintaining consistency of data over all distributed sites imposes unbearable overheads in mobile environments.

We propose a more flexible model. Semantically related or closely located data are grouped together to form a cluster. While full consistency is required for all data inside a cluster, degrees of consistency are defined for replicated data located at different clusters. The degree may depend on the availability of bandwidth in that, when the network connection is weak, users may have to tolerate higher degrees of inconsistency. The cluster configuration is dynamic. Clusters are defined or merged, as mobile users enter new cells or connect to and disconnect from the rest of the network. Consistency among clusters is restored when clusters are merged.

We augment the database interface by adding *weak operations* that have as effect the introduction of inconsistency in transactions. Users can access locally (i.e., in a cluster) consistent data by issuing weak transactions and globally consistent data by issuing strict transactions. Specifically, a weak read operation reads the locally available copy. A weak write operation writes the locally available copy. This update is later propagated to other clusters and is being committed only if does not conflict with other strict (not weak) writes.

Weak operations support disconnected operation since a mobile host can still operate as long as the user is satisfied with local copies. Furthermore, by offering to the applications the ability to specify explicitly when strict consistency is necessary for their execution, we gain in bandwidth utilization and availability. The bandwidth is also utilized better by deferring weak writes. Finally, this schema allows the modeling of operations on imprecise location data.

The above schema is explained further in [14]. In [14], we show how weak and strict transactions interact to maintain a variant of one-copy serializability, by using a variation of a 2-version method.

4 Transaction Relocation

Mobility results in an environment where the location of a user changes dynamically with time. Thus, the distance of a client from an information provider is not a fixed parameter of the cost of the service. Consequently, it may be necessary to *relocate* part of the computation that is being executed on a fixed host to another fixed host to minimize communication cost and improve response time by minimizing the physical distance between the hosts or by taking into consideration the changing network load and availability. Transaction relocation can be thought of as the dual of data relocation. Transaction relocation may also be motivated by security considerations. A base station may not be willing to support computations initiated by users that are not any more in the geographical area covered by it.

Finally, transaction relocation may be initiated to satisfy load balancing conditions among the base stations or as a result of a network or server failure. In this case, the transactions of more that one user may need to be relocated.

Upon transaction relocation, context information about the execution of the transaction must be transferred from one station to the other. This information is necessary for the new host to continue the execution of the transaction. The type and amount of this context information depends on the application and affects the performance of the system. Transaction relocation imposes a new criterion to the algorithms for maintaining the transaction properties that of supporting time and space efficient relocation. It also adds new parameters in deciding which transaction properties are desirable in mobile environments.

For instance, let T_i be a transaction that was initially submitted to site i and was later relocated to site j. We use the notation $T_{i \rightarrow i}$ for the transaction part executed at site i and $T_{i \rightarrow j}$ for the part executed at site j. $T_{i \rightarrow i}$ cannot be committed in site i. Other transactions at site i may be forced to wait for $T_{i \rightarrow i}$'s commitment that now involves remote sites. Thus, the protocols used may need redesign to provide for the conditional release of some of the local resources being held by $T_{i \rightarrow i}$. Alternatively, this may motivate the relaxation of the atomicity property of transactions. $T_{i \rightarrow j}$ inherits state information from $T_{i \rightarrow i}$. This context information depends on the consistency control method used. It may include timestamps, requested and granted locks, or log files.

We are also investigating the consequences of relocating transactions among different consistency clusters.

5 Recovery

Mobile hosts are more susceptible than static hosts to both communication and station failures. Mobile hosts are more prone to theft, loss, or accidental destruction since users carry them around with them. In addition, communication through wireless connections is more unreliable than communication through wired links.

Thus, ensuring the durability of computations performed in mobile distributed environment despite failures is a much harder task than it is in nonmobile such environments. In effect, part of the computation that is performed on a mobile host must be reported to the fixed network to achieve persistency.

To take into account the vulnerability of the computation performed at mobile hosts we introduce the concept of *transaction proxies*. For each transaction executed at a mobile host we define a dual transaction, called proxy, that will be executed on the support station of the mobile host. A proxy transaction may be considered as a subtransaction of the original transaction. Thus, any time a subtransaction is submitted to a mobile host its proxy transaction is submitted to its support station. The proxy transaction includes only the updates of the original transaction. When a host moves, a proxy may be relocated for any of the reasons described in the above section.

Alternatively, proxy transactions may be executed off-line, when the network traffic or the load in the base station is low. In that sense, proxy transactions correspond to taking periodic back-ups of the computation, which is performed at a mobile host.

We are investigating necessary criteria to schedule proxies at fixed hosts in a sound and effective way.

6 Related Research

[18] suggests that every transaction initiated by a mobile host M, is submitted to M's support station, is executed there and then the result is returned to M. Meanwhile, M can proceed independently. This model offers a practical approach. However, it ignores important issues including interactive transactions that

need input from the user and produce output, transactions that involve data stored at mobile hosts, and mobile host migration. [2] provides an axiomatic definition for two new transaction types, namely reporting transactions and co-transaction to model the interaction between a mobile host and its support station, but provides no support for weak connectivity or operation during disconnection.

File systems that support disconnected operation and weak connectivity include Coda [9, 10, 15], Ficus [5], AFS [6] and the file system in [16, 17]. All of the above systems allow operation during disconnections by adopting optimistic concurrency control methods. [11] discusses a file system schema that supports isolation-only transactions. Our work on weak transactions extends such approaches to provide support for transactions.

Consistency of replicated data is the topic of much work in database systems. We have adapted such approaches for mobile systems. For a more detailed comparison of our approach see [14].

Checkpointing of distributed applications is discussed in [1]. Our approach to recovery is based on a higher transaction level.

7 Summary

The particularities of the mobile environment make traditional transaction models inadequate. In our research, we are investigating means for providing transaction support appropriate for mobile environments. In this paper, we have provided a general discussion of the structure and properties of mobile transactions. We have defined a model that ensures a weaker notion of consistency, accounts for mobility, and provides for recovery.

References

[1] Arup Acharya, B. R. Badrinath, and T. Imielinski. Checkpointing Distributed Applications on Mobile Computing. In *Proceedings of the Third International Conference on Parallel and Distributed Information Systems*, September 1994.

[2] Panos K. Chrysanthis. Transaction Processing in Mobile Computing Environment. In *Proceedings of the IEEE Workshop on Advances in Parallel and Distributed Systems*, pages 77–83, Princeton, New Jersey, October 1993.

[3] Ahmed K. Elmagarmid, editor. *Database Transaction Models for Advanced Applications.* Morgan Kaufmann, 1992.

[4] Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques.* Morgan Kaufmann, 1993.

[5] J.S. Heidemann, T.W. Page, R.G. Guy, and G.J. Popek. Primarily disconnected operation: Experience with Ficus. In *Proceedings of the Second Workshop on the Management of Replicated Data*, November 1992.

[6] Larry B. Huston and Peter Honeyman. Disconnected operation for AFS. In *Proceedings USENIX Symposium on Mobile & Location-Independent Computing*, pages 1–10. USENIX, August 1993.

[7] Tomasz Imielinksi and B. R. Badrinath. Data Management for Mobile Computing. *SIGMOD Record*, 22(1):34–39, March 1993.

[8] Tomasz Imielinksi and B. R. Badrinath. Wireless Mobile Computing: Challenges in Data Management. *Communications of the ACM*, 1994. To appear. Also available as Technical Report, DCS-TR-296, Dept. of Comp. Sciences, Rutgers Ubiversity and WINLAB TR-49.

[9] James J. Kistler and M. Satyanarayanan. Disconnected Operation in the Coda File System. *ACM Transactions on Computer Systems*, 10(1):3–25, February 1992.

[10] Puneet Kumar and M. Satyanarayanan. Log-based directory resolution in the Coda file system. In *Proceedings of the Second International Conference on Parallel and Distributed Information Systems*, pages 202–213, 1993.

[11] Qi Lu and M. Satyanarayanan. Isolation-Only Transactions for Mobile Computing. *Operating Systems Review*, pages 81–87, April 1994.

[12] Evaggelia Pitoura and Bharat Bhargava. Dealing with Mobility: Issues and Research Challenges. Technical Report TR-93-070, Purdue University, Dept. of Comp. Sciences, 1993. Also available through the WWW, http://www.cs.purdue.edu/people/pitoura.

[13] Evaggelia Pitoura and Bharat Bhargava. Building Information Systems for Mobile Environments. In *Proceedings of the Third International Conference on Information and Knowledge Management*, Washington, DC, November 1994.

[14] Evaggelia Pitoura and Bharat Bhargava. Maintaining Consistency of Data in Mobile Distributed Environments. Technical Report TR-94-025, Purdue University, Dept. of Comp. Sciences, 1994.

[15] M. Satyanarayanan, James J. Kistler, Lily B. Mummert, Maria R. Ebling, Puneet Kumar, and Qi Lu. Experience with Disconnected Operation in a Mobile Computing Environment. In *Proceedings of the 1993 USENIX Symposium on Mobile and Location-Independent Computing*, Cambridge, MA, August 1993.

[16] Carl D. Tait and Dan Duchamp. Service Interface and Replica Management Algorithm for Mobile File System Clients. In *Proceedings of the First International Conference on Parallel and Distributed Information Systems*, pages 190–197, 1991.

[17] Carl D. Tait and Dan Duchamp. An efficient variable-consistency replicated file service. In *Proceedings of the USENIX File Systems Workshop*, pages 111–126, May 1992.

[18] L. H. Yeo and A. Zaslavsky. Submission of Transactions from Mobile Workstations in a Cooperative Multidatabase Processing Environment. In *Proceedings of the 14th International Conference on Distributed Computing Systems*, Poznan, Polland, June 1994.