# Mobile data management

Sanjay Kumar Madria
and
Sourav S. Bhowmick

© ARTVILLE/STEVE SKINNER COMPOSITE: MKC

Wireless LAN and satellite services have led to the emergence of mobile computing. As a result, users are not attached to a fixed geographical location; instead, their point of attachment to the network changes as they move. Assisted with low power, low-cost, and portable computing platforms such as laptops and personal digital assistants (PDAs), people can now work anywhere, at any time.

Mobility and portability do pose challenges to database management and distributed computing (see Fig. 1). Some problematic areas are how to handle long periods of being disconnected, limited battery life, and variable bandwidths. Also, in mobile computing, there will be more competition for shared data. Additionally, mobile users will have to share their data with others. The task of ensuring consistency of this shared data becomes more difficult in mobile computing due to the limitations and restrictions of wireless communication channels.

Some other interesting questions include: How does mobile computing differ from distributed database computing? How does mobility affect transaction processing and replication? Is location management a database management problem? And how do we replicate the location data?

## Mobile database architecture

In the mobile computing environment (see Fig. 2), the network consists of Fixed Hosts (FH), Mobile Units (MU), and Base Stations (BS) or Mobile Support Stations (MSS). Mobile units are connected to the wired network components only through base stations via wireless channels. Mobile units are battery powered, portable computers that freely move around in a restricted area called the "geographical region" (G). For example, in Fig. 2, G is the total area covered by all base stations. This cell size restriction is mainly due to the limited bandwidth of the wireless communication channels. Cell coverage is a dynamic activity that constantly changes in response to demand.

To support the mobility of mobile units and to exploit frequency reuse, the entire G is divided into smaller areas called cells. Each cell is managed by a particular base station. Each base station will store information such as a user profile, log-in files, access rights along with a user's private files. At any given instant, a mobile unit communicates only with the base station responsible for that cell area. Ultimately, a mobile unit must have unrestricted movement within G (inter-cell movement) and must be able to access the desired data from any cell.

When a mobile unit leaves a cell serviced by a base station, a hand-off protocol is used to transfer the responsibility for mobile transaction and data support to the base station of the new cell (see Fig. 3). This hand-off involves establishing a new communication link. It may also involve moving "in progress" transactions and database states without disturbing connectivity. The entire process of the handoff should be transparent to a mobile unit.

While in motion, a mobile host retains the network connections. The base stations and fixed hosts perform the transaction and data management functions with the help of a database server (DBS). This permits database processing capabilities without affecting any aspect of the generic mobile network. Data base servers can be either installed at base stations or can be a part of the fixed hosts or can be independent to both. Within this mobile computing environment, shared data are stored and controlled by a number of database servers (DBS).

Base stations provide the commonly used application software. This way a mobile user can download the software from the closest fixed host (FH) and run it on the palmtop or execute it remotely on the fixed host. Thus, the most commonly used software will be fully replicated.

A mobile unit may have some server capability to perform computations locally using a local concurrency control and recovery algorithm. Some mobile units may have very slow central processing units (CPUs) and very little memories. Thus, they act as input/output (I/O) devices only and depend on a fixed host.

- Low bandwidth
- Frequent disconnections
- High bandwidth variability
- Predictable disconnections
- Expensive
- Broadcast is physically supported in a cell
- Limited battery power
- Limited resources
- Small size and screen of laptop
- Susceptible to damaging data due to theft and accidents
- Fast changing locations
- Scalability
- Security

***Fig. 1 Constraints of mobile computing***

## Modes of operations

In a traditional distributed system, a host is either connected to the network or totally disconnected. In mobile com-

 0278-6648/01/$10.00 © 2001 IEEE 

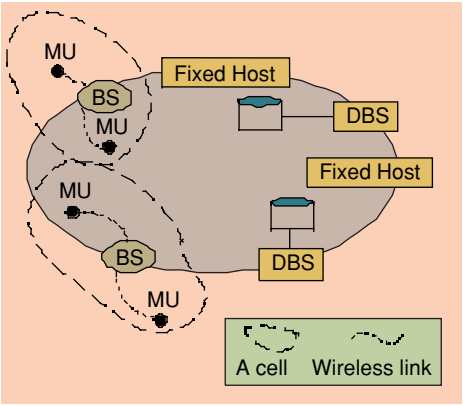


*Fig. 2 Architecture of MDS*

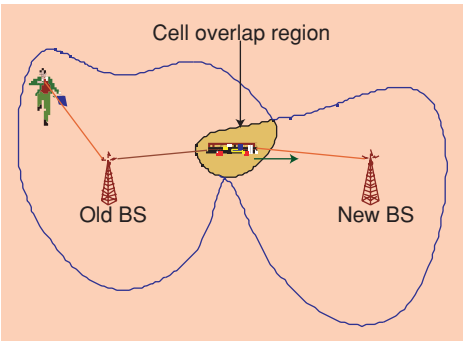


*Fig. 3 Handoff between two different cells*

puting, there are several possible modes of operations:

• fully connected (normal connection);

• totally disconnected (e.g., not a failure of mobile unit);

• partially connected or weak connection (a terminal is connected to the rest of the network via low bandwidth).

In addition to conserve energy, a mobile computer may enter an energy conservation mode called a *doze state*. In this mode, the clock speed is reduced and no user computation is performed.

These disconnected modes are usually predictable in mobile computing. Protocols can be designed to prepare the system for transitions between various modes. A mobile host should be able to operate autonomously even during total disconnection.

• A *disconnection protocol* is executed before the mobile host is physically detached from the network. The protocol should ensure that enough information is locally available (cached) to the mobile host for its autonomous operation during disconnection. It should inform the interested parties for the forthcoming disconnection.

• A *partially-disconnection protocol* prepares the mobile host for operation in a mode where all communication with the fixed network is restricted. Selective caching of data at the host site will minimise future network use.

• *Recovery protocols* re-establish the connection with the fixed network and resume normal operation.

• *Hand-off protocols* refer to the crossing the boundaries of a cell. State information pertaining to the mobile host should be transferred to the base station of the new cell.

## Mobile vs. distributed computing

A mobile computing system is a distributed system where links between nodes in the network change dynamically. Thus, we cannot rely on a fixed network structure. A single site cannot play the role of co-ordinator as in a centralized system.

The mobile host and fixed hosts also differ in computational power and memory. The distributed algorithms for mobile environments should make sure the bulk of the communication and computation costs are borne by the static portion of the network.

Also, many solutions for distributed computing problems may not work in the mobile computing arena. In a mobile environment, a database management system (DBMS) also needs to be able to recover from site, network and transaction failure, as in case of distributed systems. However, the mobility factor increases the frequency of most of these failures and complicates the recovery. Site failures at mobile units may be due to limited battery power. The mobile unit may be in *doze mode* (shutdown) which cannot be treated as a failure. Also, mobility may force more logging in to recover from failures.

Another important area is processing queries. In the mobile environment, a query may need to be distributed in at least two places. Part of a query may be executed at the mobile unit and another part may be at the fixed host using a database server.

Another issue is location-dependent query processing in a mobile environment. The same query may return different results in different locations. Here the replication of data has a different meaning than in traditional distributed database system (where all copies of a data object keep the same consistent value). In location-dependent data management, the same object in different locations may have different values but still these values are considered as consistent. For example, a "state tax" object will have different values in different states of the US.

The most important mobile computing issue remains transaction processing. Mobile database processing involves dealing with different types of disconnects, limited bandwidth and battery power and the unreliability of the communication link. Transaction failures may increase due to hand-off problems as the mobile unit moves from cell to cell. A mobile unit failure creates a partitioning of the network, which in turn complicates and affects the updating and routing algorithms.

Also, unlike a distributed transaction, a mobile transaction is not identified by a cell or by a remote site. It is identified by the collection of cells it passes through. A distributed transaction is executed concurrently on multiple processors and data sets. The execution of the distributed transaction is co-ordinated fully by the system including concurrency control, replication and atomic commit. The mobile transaction, on the other hand, is executed sequentially through multiple base stations, and on possibly multiple data sets, depending on the unit's movement. The execution of the mobile transaction is thus not fully co-ordinated by the system. Instead, the movement of the mobile unit controls the execution.

## Limited bandwidth

Mobile computing needs to be very concerned about the bandwidth consumption and variation in network bandwidth since wireless networks deliver lower variable bandwidth. Bandwidth is divided among the mobile users sharing a cell. Therefore, deliverable bandwidth per user is much lower than the raw transmission bandwidth.

Certain software techniques like compression and logging can be used for coping with low bandwidth. Data compression can be used which take less memory and communication channel but takes more CPU power to decompress. Logging can improve bandwidth usage by making large requests out of many short ones and can be combined with compression since large blocks compress better.

## Energy conservation

Energy conservation is another key issue for small palmtop units. Battery power limitations must lead to new class of "energy efficient" data access

protocols and algorithms. The following strategies can be used to deal with limited bandwidths and energy conservation:

• Data can be broadcast periodically rather than on a demand basis. There are several examples of such information include local traffic reports, stock market data and local sales events. The clients save energy by avoiding transmission and the unit wakes up from the doze mode only when absolutely necessary. Accessing broadcast data does not require up link channel and is "listen only." Many mobile hosts can listen to that broadcast, thus it supports high scalability.

• Pre-fetching can be used to download the files before they are needed.

• It is better to perform the execution at the fixed server rather than at the mobile client. Hence, for a given amount of energy, the trade-off is between the amount of data that can be accessed locally and the amount of data that will be processed on request remotely and delivered later. This however requires data to be partitioned between the client and the server. Another factor is the processing speed. Here again, the longer the latency that can be tolerated in processing, the less energy that is consumed.

• The ability to operate disconnected can be useful even when connectivity is available. For example, disconnected operations can extend battery life by avoiding wireless transmission and reception.

## Reliability of communication

Wireless connections are of lower quality due to lower bandwidth, higher error rates and more frequent disconnects. These factors together can increase the communication downtime and cost due to retransmission, time-out delays and error control protocol processing.

Wireless connections can also be lost due to mobility. Users may enter areas of high interference or large concentration such as conventions or other public events that may result in overloading the network's capacity. But some of these problems are foreseeable. A user may be able to pre-announce future disconnection from the network or power down of the computer. Changing signal strength in a wireless network may allow the system to predict an imminent disconnection.

Foreseeable disconnects imply that the system should be able to take special action on behalf of active transactions at the time a disconnection is predicted. These include:

• Transaction process may be migrated to a non-mobile computer if no further user interaction is needed.

• Remote data may be downloaded in advance of the predicted disconnection to support interactive transactions that should continue to execute locally on the mobile machine after disconnection.

• Log records may be transferred from the mobile computer to a non-mobile computer. This is particularly important because of the instability of storage in mobile computing. Highly reliable systems use replicated logs since a mobile computer is uniquely vulnerable to a catastrophically failure due to user dropping the machine, data distraction by an airport security system, or even the loss or theft of the entire machine.

• The mobile computer may take action to "declare itself down" by removing itself from quorums for distributed protocol to handle the disconnection with less overhead than in current models in which disconnection is only discovered only after it occurs.

## Mobile data management

Data management in mobile computing can be described as global and local data management. Global data management deals with network level issues such as location, addressing, replication, broadcasting, etc. Local data management refers to the end user level that includes energy efficient data access, management of disconnection and query processing.

## Location data management

A mobile user's location is of prime importance in wireless computing. This is because the location of a user can be regarded as a data item whose value changes with every move. Thus, location management is a data management problem. Primary issues here are how do we know the current position of the mobile unit? Where do we store the location information, and who should be responsible for determining and updating the information?

To locate users, distributed location databases are deployed that maintain the current location of mobile users. Thus, location data can be treated as a piece of data that is updated and queried. The search for this data should be as efficient as any other queried data.

Writing the location variable may involve updating the location of the user in the location database as well as in other replicated databases. The location management involves searching, reading, informing and updating. If A wants to find the location of B, should A search the whole network or only look at pre-defined locations? Should B inform any one before relocating?

Management assumes that each user is attached to a home location server (or home location register (HLR)) that always "knows" the unit's current address. When a user moves, the home location server is informed about the new address. To send a message to such a user, the person's home location register is contacted first to obtain the current address. A special form of "address embedding" is used to redirect the packets addressed to the mobile user from the home location to the current location. This scheme works well for the user who stays within his or her home area, it does not work for global moves.

In this algorithm, when a user A calls user B, the lookup algorithm initiates a remote lookup query to the home location register of B. However, this may be at a remote site. And performing remote queries can be slow due to high network latency.

An improvement over such algorithm is to maintain visitor location registers (VLR). The visitor location register at a geographical area stores the profiles of users currently only visiting that area temporarily. If a user's profile is not found, then it queries the database in the user's home area. This technique is useful in the case when a user receives many calls while visiting an area since it avoids queries to the home location register while at a remote site.

Visitor location registers can be viewed as a limited replication scheme. Each user's profile is located in its current area when the person is not in his or her home area.

## Cache consistency

Caching of frequently accessed data plays an important role in mobile computing. Caching is useful during frequent relocation and connection to different database servers. Caching of frequently accessed data items will reduce contention on the small bandwidth wireless network. This will improve query response time, and help to support disconnected or weakly connected operations.

If a mobile user has cached a portion of the shared data, the person may

request different levels of cache consistency. In a strongly connected mode, the user may want the current values of the database items belonging to the cache. During weak connections, the user may require weak consistency when the cached copy is a quasi-copy of the database items. Each type of connection may have a different degree of cache consistency associated with it. That is, weak connection corresponds to "weaker" level of consistency.

Cache consistency is severely hampered by both the disconnection and mobility of clients since a server may be unaware of the current locations and connection status of clients. This problem can be solved by the server periodically broadcasting either the actual data, an invalidation report (reports the data items which have been changed), or even control information such as lock tables or logs.

Broadcasting is attractive since the server need not know the location and connection status of its clients and the clients need not establish an up link connection to a server to invalidate their caches. Also, the mobile host saves energy since it need not transmit data requests, and many mobile hosts can receive the data with no extra cost.

Depending upon what is broadcast, an appropriate scheme can be developed. Given the rate of updates, the trade-off is between the periodicity of broadcast and the divergence of the cached copies that can be tolerated. The more inconsistency that can be tolerated the less often the updates need to be broadcast.

Given a query, the mobile host may optimize energy costs by determining whether it can process the query using cached data or transmit a request for data. Another choice could be to wait for the relevant broadcast.

However, cache coherence preservation under weak-connections is expensive. Large communication delays increase the cost of validating cached objects. Unexpected failures increase the frequency of validation since it must be performed each time communication is restored. An approach that only validates on demand could reduce validation frequency. But this approach would lessen the consistency since some old objects could be accessed while disconnected.

## Data replication

The ability to replicate the data objects is essential in mobile computing to increase availability and performance. Shared data items have different synchronization constraints depending on their semantics and particular use. These constraints should be enforced on an individual basis. Replicated systems need to provide support for the disconnected mode, data divergence, application defined reconciliation procedures, optimistic concurrency control and so forth.

Replication also permits the system to ensure transparency for mobile users. A user who has relocated and has been using certain files and services at the previous location wants to have the same environment recreated at the new location. Mobility of users and services and its impact on data replication and migration still needs to be resolved. There are many issues:

• How to manage data replication, providing the levels of consistency, durability and availability needed.

• How to locate objects of interest. Should information about location also be replicated and to what extent (is the location dynamically changing the data item)?

• Under what conditions do we need to replicate the data on a mobile site?

• How does the users' moves affect the replication scheme? How should the copy follow the user? In general, should data move closer to the user?

• Does the mobile environment require dynamic replication schemes?

• Do we need new replication algorithms or can we just modify the proposed replication schemes for distributed environment?

## Mobile transaction processing

A transaction in a mobile environment is different from a transaction in a centralized or a distributed database in the following ways:

• The mobile transaction might have to split its computations into sets of operations, some that execute on the mobile host while others execute on the stationary host. A mobile transaction shares its states and partial results with other transactions due to possible disconnection and/or movement to another cell.

• The mobile transaction requires computations and communications to be supported by stationary hosts.

• When the mobile user moves during the execution of a transaction, it continues its execution in the new cell. The partially executed transaction may be continued at the fixed local host according to the instruction given by the mobile user. Different mechanisms are required if the user wants to continue its transaction at a new destination.

• As the mobile hosts move from one cell to another, the states of the transaction, states of the accessed data objects, and the location information also move.

• The mobile transactions are long-lived transactions due to the mobility of both the data and users, and due to the frequent disconnects.

• The mobile transaction should support and handle concurrency, recovery, disconnection and mutual consistency of the replicated data objects.

• The transaction processing models should accommodate the limitations of mobile computing, such as unreliable communication, limited battery life, low bandwidth communication and reduced storage capacity.

Mobile computations should minimize aborts due to disconnection. Operations on shared data must ensure correctness of transactions executed on both stationary and mobile hosts. The blocking of a transaction's executions on either the stationary or mobile hosts must be minimized to reduce communication cost and to increase concurrency. Proper support for mobile transactions must provide for local autonomy to allow transactions to be processed and committed on the mobile host despite temporary disconnection.

In optimistic concurrency control based schemes, cached objects on mobile hosts can be updated without any co-ordination. But the updates need to be propagated and validated at the database servers. This scheme leads to aborts of mobile transactions unless the conflicts are rare.

In pessimistic schemes in which cached objects can be locked exclusively, mobile transactions can be done locally. The pessimistic schemes lead to unnecessary transaction blocking since a mobile host cannot release any cached objects while it is disconnected. Existing caching methods attempt to cache the entire data objects or, in some cases, the complete file. Caching these potentially large objects over low-bandwidth communication channels can result in wireless network congestion and high communication cost. The limited memory size of the mobile unit

allows only a small number of objects to be cached at any given time.

## Location-dependent query processing

Processing that deals with location-dependent data can be a subject of more complex aggregate queries. For example, finding the number of hotels in the area you are currently in or looking for a doctor closest to your present location. Hence, the location information is a frequently changing piece of data. The objective is getting the right data at each different location to process a given query. The results provided should satisfy the location constraints with respect to the point of query origin, where the results are received, etc.

We propose building additional capabilities into the existing database systems to handle location-dependent data and queries. Data may represent the social security number (SSN) of a person or the sales tax of a city. In one representation, the mapping of the data value and the object it represents is not subjected to any location constraints.

For example, the value of a person's SSN remains the same no matter from which location it is accessed. This is not true for sales tax data. The value of the sales tax depends on where the query is executed. For example, sales tax value of West Lafayette is governed by a different set of criteria than the sales tax of Boston. We can, thus, identify data whose value depends on criteria established by the location and data not subject to the constraints of a location.

There is a third type of data that is sensitive to the point of query. Consider a commuter who is travelling in a taxi and initiates a query on his laptop to find nearby hotels. The answer to this query depends on the location of the origin of the query. Since the commuter is moving he may receive different results at a different location.

Thus, the query results should correspond to the location where the result is received or to the point of the origin of the query. The difference in the two correct answers to the query depends on the location and not on the hotel. The answer to the query "find the cheapest hotel" is not affected by the movement.

In a project called MOST, a database is considered that represents information about moving objects and their location. The project's backers argue that existing database management systems are not well equipped to handle continuously changing data, such as location of moving objects. They address the issue of location modelling by introducing the concept of dynamic attribute (whose value keeps changing), spatial and temporal query languages and indexing dynamic attributes.

## Conclusions

Management of data in the mobile computing environment offers new challenging problems. Existing software needs to be upgraded to accommodate this environment. To do so, the critical parameters need to be understood and defined. We have surveyed some problems and existing solutions. There is a need to explore these issues further and improve the existing solutions offered.

## Read more about it


• B. Daniel, "Mobile computing and databases: A survey," *IEEE Trans. Knowledge Data Eng.*, 1999.

• B. Bruegge and B. Benninington, "Applications of mobile computing and communications," *IEEE Pers. Commun.,* vol. 3, Feb., 1996.

• J. Cai, K.L. Tan, and B.C. Ooi, "On incremental cache coherency schemes in mobile computing environment," in *Proc. 1997 IEEE Int. Conf. Data Engineering.*

• Y. Huang, P. Sistla, and O. Wolfson, "Data replication for mobile computers," in *Proc. 1994 ACM SIGMOD Int. Conf. on Management of Data.*

• T. Imielinksi and B.R. Badrinath, "Wireless mobile computing: Challenges in data management," *Communications of ACM*, vol. 37, no. 10, Oct. 1994.

• P. Krishna, N.H. Vaidya, and D.K. Pradhan, "Static and dynamic location management in mobile wireless networks," *Journal of Computer Communications* (special issue on Mobile Computing), vol. 19, no. 4, Mar. 1996.

• Y. Lin, "Reducing location update cost in PCS networks," *IEEE/ACM Trans. Networking,* vol. 5, no. 1, pp. 25-33, 1997.

• S.K. Madria, M. Mohania, B. Bhargava, and S. Bhowmick, "A study on mobile data and transactions," accepted *Information Science Journal*, 2001.

• E. Pitoura and I. Fudos, "An efficient hierarchical scheme for locating highly mobile users," in *1998 ACM Proc. Int. Conf. Information and Knowledge Management.*

• M. Satyanarayanan, "Mobile information access," *IEEE Pers. Commun.,* vol. 3, Feb., 1996.

• O. Wolfson and S. Jajodia, "Distributed algorithms for dynamic replication of data," in *Proc. Symp. Principles of Database Systems,* CA, 1992, pp. 149-163.


## About the authors


Sanjay Kumar Madria received his Ph.D. in Computer Science from the Indian Institute of Technology in Delhi, India in 1995. He is an Assistant Professor with the Department of Computer Science at the University of Missouri-Rolla. Prior to that he was a visiting Assistant Professor with the Department of Computer Science at Purdue University in West Lafayette, IN. He has also held appointments at Nanyang Technological University in Singapore and University Sains Malaysia in Malaysia. He has published more than 50 papers in the areas of Web warehousing, mobile databases, data warehousing, nested transaction management and performance issues. He guest edited WWW Journal and Data and Knowledge Engineering for special issues on Web data management and data warehousing. He was the Program Chair for the EC&WEB 2001 conference held in Germany in September, 2001. He is serving as a Program Chair member at various database conferences and workshops and is a reviewer for many reputable database journals. Dr. Madria has given tutorials on Web warehousing and mobile databases at many international conferences.

Sourav S. Bhowmick is currently an Assistant Professor at Nanyang Technological University in Singapore. His current research interests include XML data management, change management on the Web, Web warehousing, Web mining, and mobile location-sensitive data. He earned his Ph.D. degree from Nanyang Technological University in 2001 and received his Master's degree in Computing from Griffith University in Australia. He has published 17 conference and journal papers for various international conferences and international journals including Data and Knowledge Engineering, World Wide Web Journal, and Computer Journal, among others.