Mobile data and transaction management

Sanjay Kumar Madria [a,*], Mukesh Mohania [b], Sourav S. Bhowmick [c], Bharat Bhargava [d]

[a] *Department of Computer Science, University of Missouri-Rolla, Rolla MO, USA*
[b] *IBM India Research Lab, Block No 1, IIT, Hauz Khas, New Delhi, India*
[0] *School of Computer Engineering, Nanyang Technological University, Singapore*
[d] *Department of Computer Science, Purdue University, West Lafayette, IN, USA*

Abstract

Mobile computing paradigm has emerged due to advances in wireless or cellular networking technology. This rapidly expanding technology poses many challenging research problems in the area of mobile database systems. The mobile users can access information independent of their physical location through wireless connections. However, accessing and manipulating information without restricting users to specific locations complicates data processing activities. There are computing constraints that make mobile database processing different from the wired distributed database computing. In this paper, we survey the fundamental research challenges particular to mobile database computing, review some of the proposed solutions and identify some of the upcoming research challenges. We discuss interesting research areas, which include mobile location data management, transaction processing and broadcast, cache management and replication and query processing. We highlight new upcoming research directions in mobile digital library, mobile data warehousing, mobile workflow and mobile web and e-commerce.

Keywords: Mobile computing; Wireless or cellular networking; Mobile database; Distributed database

1. Introduction

The rapid technological advancements in cellular communications, wireless LAN and satellite services have led to the emergence of mobile computing [11]. In mobile computing, users are not attached to a fixed geographical location; instead their point of attachment to the network changes as they move. The emergence of relatively sophisticated low-power, low-cost and portable computing platforms such as laptops and personal digital assistants (PDA) have made possible for people to work from anywhere at any time (from their offices, homes and while travelling) via wireless communication network. As the technology advancing, millions of users carry portable computer and communicator devices that use a wireless connection to access worldwide global information network. Each mobile unit equipped with wireless network can be connected to global information network to provide unrestricted user mobility.

Mobility and portability pose new challenges to the mobile database management and distributed computing [34]. The database software support for mobile computing is still in the germinating stage. There is necessity to design specifications for energy efficient data access methodologies and in general develop database software systems that extend existing database systems designs and platforms to satisfy the constraints imposed by mobile computing (see Fig. 1). How to handle long period of disconnection, and other constrained resources of mobile computing such as limited battery life and variable bandwidth etc.? In mobile computing, there will be more competition for shared data since it provides users with ability to access information and

- **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.

services through wireless connections that can be retained even while the user is moving. Further, mobile users will have to share their data with others. The task of ensuring consistency of share data becomes more difficult in mobile computing because of limitations and restrictions of wireless communication channels.

Some of the several questions need to be answered are how mobile computing differs from distributed database computing? How does mobility affect transaction processing and replication? Is location management a database management problem? How to replicate the location data? Caching is key to handle frequent disconnection in mobile computing. How to keep cache consistent with minimum communication cost? Is the weaker notion of consistency more appropriate? How query processing is different in mobile computing environment?

In this paper, we discuss some of the problems identified with mobile database computing and review proposed solutions, and explore the upcoming research challenges. Some of the problems involved in supporting transaction services and distributed data management in a mobile environment has been identified in [2,8,34,46]. At the time of submitting this paper, we found a recent survey, which is very specific towards data dissemination, location-dependent query and advanced interfaces for mobile computers in [8]. Our survey here is much more comprehensive and gives elaborate coverage of wide variety of research issues in mobile data management, not reported in [8]. Also, we categorize mobile database research and discuss future research issues and applications on mobile platform, again not covered in [8].

The rest of the paper is organized as follows. Section 2 discusses the mobile database architecture. We highlight data processing and mobile constraints issues in Section 3. In Section 4, we give in depth treatment to various the mobile data management issues. Section 5 discusses transaction processing in mobile databases. In Section 6, we explore upcoming mobile database research directions. We conclude in Section 7.

2. Mobile database architecture

In mobile computing environment (see Fig. 2), the network consists of fixed hosts (FHs), mobile units (MUs) and base stations (BSs) or mobile support stations (MSS). MUs are connected to the wired network components only through BS via wireless channels. MUs are battery powered portable computers, which move around freely in a restricted area, which we refer to as the "geographical region" (G). For example in Fig. 2, *G* is the total area covered by all BSs. This cell size restriction is mainly due to the limited bandwidth of wireless communication channels. To support the mobility of MUs and to exploit frequency reuse, the entire G is divided into smaller areas called cells. A
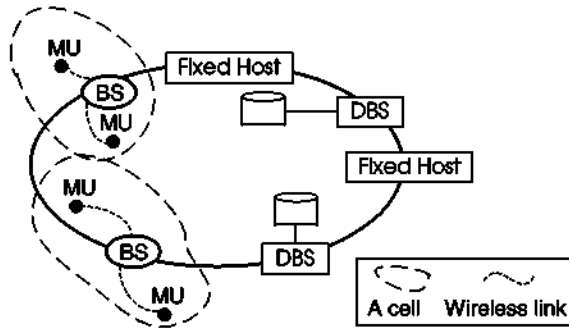
Fig. 2. Architecture of MDS.

particular BS manages each cell. Each BS will store information such as user profile, login files, and access rights together with user's private files. At any given instant, an MU communicates only with the BS responsible for its cell. The mobile discipline requires that an MU must have unrestricted movement within *G* (inter-cell movement) and must be able to access desired data from any cell.

An MU changes its location and network connections while computations are being processed. While in motion, a mobile host retains its network connections through the support of BSs with wireless connections. The BSs and FHs (fixed hosts) perform the transaction and data management functions with the help of database server (DBS) component to incorporate database processing capability without affecting any aspect of the generic mobile network. DBSs can either be installed at BSs or can be a part of FHs or can be independent to BS or FH.

BSs will provide commonly used application software so that a mobile user can download the software from the closest FH and run it on the palmtop or execute it remotely on the FH. Thus, the most commonly used software will be fully replicated. A mobile host may play a different role in a distributed system. An MU may have some server capability to perform computations locally using local concurrency control and recovery algorithm. Some MU may have very slow CPU and very little memory and thus, acts as an I/O device only. Thus, they depend on some FHs. Within this mobile computing environment, shared data are stored and controlled by a number of DBSs.

When an MU leaves a cell serviced by a BS, a hand-off protocol is used to transfer the responsibility for mobile transaction and data support to the BS of the new cell. This hand-off involves establishing a new communication link. It may also involve migration of in progress transactions and database states from one BS to another. The entire process of handoff is transparent to an MU and is responsible for maintaining end-to-end data movement connectivity.

The wireless interface can be either a cellular network with bandwidth of 10-20 Kbps or a wireless local area network (LAN) with 10 Mbps of bandwidth (e.g., NCR Wavelan, Motorola ALTAIR). The fixed wired network on the other hand can offer bandwidth of 10 Mbps for Ethernet, up to 100 Mbps for FDDI and 144 Mbps for ATMs.

2.1. Modes of operations

In mobile computing, there are several possible modes of operations [56,57] than in a traditional distributed system where a host may operate only in one of two modes; either connected to the network or totally disconnected. The operation mode in mobile computing may be one of the following:
- fully connected (normal connection);
- totally disconnected (e.g., not a failure of MU);
- partially connected or weak connection (a terminal is connected to the rest of the network via low bandwidth).

In addition, for conserving energy, a mobile computer may also enter an energy conservation mode, called *doze state* [56]. A doze state of MU does not imply the failure of the disconnected machine. In this mode, the clock speed is reduced and no user computation is performed.

The most of these disconnected modes are 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 minimize future network use.
- *Recovery protocols* re-establish the connection with the fixed network and resume normal operation.
- *Hand-offprotocols* 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.

3. Mobile computing verses distributed computing

A mobile computing system is a dynamic type of 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 FHs also differ in computational power and memory. The distributed algorithms for mobile environments should be structured such as that the main bulk of the communication and computation costs is borne by the static portion of the network. In [10], they introduced the idea of associating with each mobile host a proxy on the static network, thus decoupling mobility from the design of the algorithm.

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

Caching at MU is an interesting idea to optimize the use of wireless connections by increasing availability. Its application in the WWW environment is very useful where size of data in enormous. However, maintaining cache consistency is an important objective and different consistency requirement can be forced depending on the applications. Catch needs to be updated frequently and thus, needs new update protocols. Replication in mobile environment certainly increases availability but may need certain weaker consistency criteria [57]. Also, replication schemes for distributed systems may not directly applicable here and there is a need for dynamic replication schemes [81].

Another important area is query processing. In mobile environment, query may need to be distributed at least at two places. Part of query may be executed at MU and other part may be at FH with the help of DBS. Another interesting issue is location-dependent query processing in mobile environment where query returns results according to the location [51,83]. Thus, same query may return different results in different locations. Here the replication of data has a different meaning than in traditional distributed database where all copies of data object keeps same consistent values. In location-dependent data management, same object in different locations may have different values but still these values are considered as consistent. For example, tax object have different values in different states in United States.

The most important issue remains is transaction processing in such an environment. Transaction failures may increase due to the possibility of problem during hand-off when the MU moves between cells. An MU failure creates a partitioning of the network, which in turn complicates updating and routing algorithms. Another major difference lies in the transaction model. Unlike a distributed transaction, a mobile transaction is not identified by a cell or 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 movement of the MU. The execution of the mobile transaction is thus not fully co-ordinated by the system. The movement of the MU controls the execution.

We will review some of the work in the above-mentioned areas in this paper.

4. Data processing and mobile constraints

The important characteristics of mobile databases processing involves dealing with different types of disconnections, limited bandwidth and battery power and the unreliability of communication link. In this section, we focus on many of such issues.

4.1. Limited bandwidth and energy efficient data management

Mobile computing needs to be more 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. Energy conservation is another key issue for small palmtop units. There is growing demand for energy efficient CPUs, memories as well as system software. 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 [1] rather than provide on demand basis. There are several examples of information such as local traffic information, stock market data. local sales, events, news that would rather be broadcasted than provide 'on demand' basis. The clients save energy by avoiding transmission and waking 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.
- Certain software techniques like compression, logging can also 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.
- 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, longer the latency that can be tolerated in processing, lower the energy consumed.
- The ability to operate disconnected can be useful even when connectivity is available. For example, disconnected operation can extend battery life by avoiding wireless transmission and reception.

Thus, processing speed, storage cost, and amount of data transmitted and received and the tolerable latency will be factors in considering various data access and data organization.

4.2. Reliability of communication

Wireless connections are of lower quality due to lower bandwidth, higher error rates, and more frequent disconnections. These factors together can increase the communication latency and cost due to retransmission, time-out delays, error control protocol processing, and short disconnections. Wireless connections can also be lost due to mobility. Users may enter areas of high interference or large concentration such as conventions, public events etc. which may result in overloading of network capacity. These factors suggest that the mobile environment is more failure prone than the traditional one but some of these 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 imminent disconnection.

Foreseeable disconnection of mobile computers implies that the system must be able to take special action on behalf of active transactions at the time a disconnection is predicted.

- 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 fewer overheads than in current models in which disconnection is only discovered only after it occurs.

5. Mobile data management

In this section, we will discuss some of the important data management issues with respect to mobile computing. 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.

5.1. Location data management

The location of mobile user [64] is of prime importance in wireless computing. In a mobile computing, the location of a user can be regarded as a data item whose value changes with every move. In the mobile computing, the location management is a data management problem. Primary issues here are how to know the current position of the MU? Where to store the location information and who should be responsible for determining and updating of information? To locate users, distributed location databases are deployed which maintain the current location of mobile users. The location data can be treated as a piece of data that is updated and queried. The search of this piece of 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. One such method is described in [36]. It assumes that each user is attached to a home location server (now generally referred as home location register (HLR)) that always "knows" his current address. When a user moves, he informs his home location server about his new address. To send a message to such a user, his **HLR** is contacted first to obtain his current address. A special form of 'address embedding' is used to redirect the packets addressed to the mobile user from the home location to his current location. This scheme works well for the user who stay within their respective home areas, 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 **HLR** of B, which may be at remote site. Performing remote queries can be slow due to high network latency. An improvement over such algorithm [52] is to maintain visitor location registers **(VLR).** The **VLR**

at geographical area stores the profiles of users currently located in that area for whom the area is not their home. The query then calls in caller's area and if the callee's profile is not found, it queries the database in callee's home area. This is useful when a callee received many calls from users in the area he is visiting since it avoids queries to HLR of callee at remote site. VLR's can be viewed as limited replication scheme since each user's profile is located in its current area when he is not in his home area. Another scheme proposed in [28], handles global moves on the assumption that most messages are exchanged between parties or between user in a remote area and its home location area.

Awerbuch and Peleg [5] consider a formal model for on-line tracking of users by decomposing PCS (personal communication system) network into regions and using regional directories. They discuss how to trade-off search and update costs while tracking users. Badrinath and others [15,19] propose per-user placement which uses cell partitions where user travels frequently and separating the cells between which it relocates infrequently to control network traffic generated by frequent updates. Only moves which are across the partitions, are reported.

In [73], the location lookup problem is considered to find callee within the reasonable time bounds to set up the call from the caller to callee. Each user is located in some geographical area where the mobile service station keeps track of each user in the form of hPID,ZID) where PID and ZID uniquely identified the MU id and its current location id, respectively. They replicate per-user profile based on calling and mobility patterns. Thus, they balance the storage and update costs and at the same time provide fast lookups. The decision where to replicate the profiles is based on a minimum-cost maximum-flow [7] algorithm. They maintain an up-to-date copy of a user profile at the user's HLR and in addition, they also find out the sites at which a user's profile will be replicated. Thus, the algorithm does not guarantee that a user's profile will be found in his current area.

Jain et al. [39] propose per-user caching where geographical areas cache the last known location of certain users for fast lookup. Replication of profiles keep all copies up-to-date where as caching may not have up-to-date user profile.

Hierarchical distributed database architectures [6,37,43,70] have been build to accommodate the increased traffic associated with locating moving users. In these models, each leaf database covers a specific geographical region and maintains location information for all users currently residing in that region. Location databases at internal nodes contain information about the location of all the users registered in areas covered by the databases at their children nodes. A hierarchical method for location binding in wide-area system is used in the Globe wide-area location service [70]. Globe uses a combination of caching and partitioning.

Anantharaman et al. [6] discuss the assignment of location databases to the nodes of a signalling network. It uses dynamic programming to optimize the mapping, which maps a database hierarchy to the network configuration, based on fixed calling and mobility patterns. However, they do not consider communication costs and do not adapt to change in patterns.

Dolev and Pradhan [23] again use a tree based structure for location database. They modify the structure to balance the average load of search requests by replacing the root and some of the higher levels of the tree with set-ary butterfly (a generalization of K-ary butterfly). They modify the lowest level of the tree to reflect neighboring geographical regions more accurately and to allow simple hand-offs.

In [31], the hierarchical scheme allows dynamic adjustment of the user location information distribution based on mobility patterns of MUs. A unique distribution strategy is determined for each mobile terminal and location pointers are set up at selected remote locations. This reduces database access overhead for registration and there is no need for centralized co-ordination.

Forwarding pointers have been used in hierarchical location databases [43,62]. In [62], the objective is to reduce the cost of moves by updating only databases up to a specific level of the tree and a forwarding pointer is set at lower level in the database. However, if forwarding pointers are never deleted, then long chains are created, whose traversal results in increase in the cost of locating users during calls. They introduced caching techniques that reduce the number of forwarding pointers to travel before locating the calling as well as conditions for initiating a full update of the database entries. They have also described a synchronization method to control the concurrent execution of call and move operations. The difference between the two schemes [43,62] is that in [43] actual location is saved at each internal level database instead of pointer to the corresponding lower level database. The forwarding pointers are set at different level in the hierarchy and not necessarily at the lower level database as in [62]. In [43], the objective is to choose an appropriate level for setting the forwarding pointers and on updating obsolete entries in the hierarchy after a successful call. Jain [37] proposes caching in hierarchical structures instead of replication to reduce the cost of calls. Some more results on optimum location management algorithms have appeared in [41,44].

5.2. Cache consistency

Caching of frequently accessed data plays an important role in mobile computing because of its ability to alleviate the performance and availability limitations during weak-connections and disconnections. Caching is useful during frequent relocation and connection to different DBS. In a wireless computing, caching of frequently accessed data items is an important technique that will reduce contention on the small bandwidth wireless network. This will

improve query response time, and to support disconnected or weakly connected operations. If a mobile user has cached a portion of the shared data, he 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 his 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. The server can solve this problem by periodically broadcasting the actual data, invalidation report (reports the data items which have been changed), or even control information such as lock tables or logs. This approach is attractive in mobile environments since the server need not know the location and connection status of its clients and clients need not establish an up link connection to a server to invalidate their caches. There are two advantages of broadcasting. First, mobile host saves energy since they need not transmit data requests and second, broadcast data can be received by many mobile hosts at once with no extra cost.

Depending upon what is broadcasted, the appropriate schemes can be developed for maintaining consistency of data of a distributed system with mobile client. Given the rate of updates, the trade-off is between the periodicity of broadcast and divergence of the cached copies that can be tolerated. The more the inconsistency tolerated the less often the updates need to be broadcasted. 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.

Cache coherence preservation under weak-connections is expensive. Large communication delay increases the cost of validation of 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 worsen consistency since it increases the possibility of some old objects being accesses while disconnected.

In Coda [68], during the disconnected operation, a client continues to have read and write access to data in its cache. Coda file system allows the cached objects in the mobile host to be updated without any co-ordination. When connectivity is restored, the system propagates the modifications and detects update conflicts. The central idea is that caching of data and the key mechanisms for supporting disconnected operation which includes three states: hoarding, emulation and reintegration. The client cache manager while in hoarding state relies on server replication, but is always on the alert for possible disconnection and ensures that critical objects are cached at the time of

disconnection. Upon disconnection, it enters the emulation state and relies solely on the contents of the cache. Coda's original technique for cache coherence while connected was based on callbacks [71]. In this technique, a server remembers that a client has cached an object, and promises to notify it when the object is updated by another client. This promise is called callback, and the invalidation message is a callback break. When a callback break is received, the client discards the cached copy and refetches it on demand. When a client is disconnected, it can no longer rely on callbacks. Upon reconnection, it must revalidate all cached objects before use to detect updates at the server.

Cache invalidation strategies will be affected by the disconnection and mobility of clients. The server may not have information about the live MUs in its cells. Barbara and Imielinski [14] propose taxonomy of different cache invalidation schemes and study the impact of client's disconnection times on their performance. They address the issue of relaxing consistency of caches. They use quasi-copies whose values can deviate in a controlled manner. They have categorized the MUs on the basis of the amount of time they spend in their sleep mode into sleepers, and workaholics. Different caching schemes turn out to be effective for different populations. Broadcast with timestamps are proved to be advantageous for frequent queries than the rate of updates provided that units are not workaholics.

Wu et al. [84] propose a technique to decide whether some items in the cache can still be used by the MU even after it is connected to the server. The database is partitioned in different groups and items in the same group are cached together to decrease the traffic. Thus, MU has to invalidate only the group rather than individual items. In [24], various alternative caching strategies for mobile computing have been evaluated. More work needs to be done in the direction of performance evaluation and availability limitation of various caching under weakly connected and disconnected operation.

In [20], an incremental cache coherency problem in mobile computing is examined in the context of relational operations select, project and join. Taxonomy of cache coherency schemes is proposed as case studies. However, it does not address the problems of query processing and optimization and does not include other relational operations.

5.3. 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 disconnected mode, data divergence, application defined reconciliation procedures, optimistic concurrency control, etc. Replication is a way by which the system ensures transparency for mobile

users. A user who has relocated and has been using certain files and services at the previous location wants to have his environment recreated at the new location. Mobility of users and services and its impact on data replication and migration will be one of the main technical problems to be resolved. There are many issues raised by the relocated data and mobility of users and services:

- How to manage data replication, providing the levels of consistency, durability and availability needed.
- How to locate objects of interest. Should information about location be also replicated and to what extent (location is dynamically changing data item).
- What are the conditions under which we need to replicate the data on a mobile site.
- How users' moves affect the replication scheme. How should the copy follow the user. In general data should move closer to the user.
- Is mobile environment requiring dynamic replication schemes [81].
- Do we need new replication algorithms or the proposed replication schemes for distributed environment can be modified.

In [12], caching of data in mobile hosts and the cost of maintaining consistency among replicated data copies have been discussed. It allows caching of data to take place anywhere along the path between mobile/fixed servers and clients. It determines via simulations which caching policy best suits given mobility and read/write patterns.

Ravindran and Shah [66] consider a general model for maintaining consistency of replicated data in distributed applications. It defines a casualty constraint, a partial ordering between application operations, such that data sharing is achieved by defining groups requiring it and broadcasting updates to the group. Each node processes the data according to the constraints.

In [27], It has been argued that traditional replica control methods are not suitable for mobile databases and the authors have presented a virtual primary copy method. In this method, the replica control method decides on a transaction-by-transaction basis whether to execute that transaction on mobile hosts primary copy or virtual primary copy. This method requires a transaction to be restarted when mobile host disconnects. Also, when a mobile host reconnects, it either has to wait for the completion of all transactions executed on virtual copy before synchronizing itself with the rest of the system or all running transactions will have to be restarted.

Huang et al. [32] present an analysis of various static and dynamic data allocation methods with the objective of optimizing the communication cost between a mobile computer and the stationary computer that stores on-line database. They consider one-copy and two copies allocation schemes. In static scheme, allocation scheme remain unchanged where as in dynamic scheme allocation method changes based on the number of reads and writes. If in the last k requests there are more reads at MU than writes at stationary computer, it uses two copy scheme. Otherwise it uses one-copy schemes. Two cost

models were developed for cellular phones (user is charged per minute of connection) and packet radio networks (user is charged per message basis), respectively.

A new two-tier replication algorithm is proposed by Gray et al. [30] to alleviate the unstable behavior observed in the update anywhere-anytime-anyway transactional replication scheme when the workload scales up. Lazy master replication that is employed in the algorithm assigns an owner to each object. The owner stores the object's correct value. Updates are first done by the owner and then propagated to other replicas. The two tier scheme uses two kinds of nodes: mobile nodes (may be disconnected) and base nodes (always connected). The mobile nodes accumulate tentative transactions that run against the tentative database stored at the node. Each object is mastered at either the mobile node or the base node. When the mobile node reconnects to the base station, it sends replica updates mastered at the mobile node, the tentative transactions and their input parameters to the base node. They are to be re-executed as base transactions on the master version of data objects maintained at the base node in the order in which they are committed on the mobile node. If the base transaction fails its acceptance criterion, the base transaction is aborted and a message is returned to the user of the mobile node. While the transaction executed on the objects mastered on the mobile nodes are confirmed, those executed on the tentative objects have to be checked with nodes that hold the master version.

A dynamic replication scheme which employs user profiles for recording users' mobility pattern, access behavior and read/write patterns, and actively reconfigures the replicas to adjust to the changes in the user locations and systems is proposed in [80]. They devise the concept of open objects to represent a user's current and near future data requirements. This leads to more precise and responsive cost model to reflect changes in access patterns.

5.4. Query processing

Query processing in mobile computing environment involves two types of queries. First type of query may involve only the content of databases. Another type may involve the queries which may include location-dependent data and furthermore data that depend on the direction of movement. Thus, queries may introduce new parameters regarding query optimization. Location data may change during query evaluation. Queries may be answered in an approximate way due to fast changing location data. How to keep track of the value of the query involving broadcasted data in constantly changing environments? Another issue is querying the broadcasted data. What is the best execution plan for a query that involves data broadcasted on different channels? What should be the organization of the broadcasted data so that the energy spent on the client's side is minimized. Which information should be broadcasted and which

should be provided on demand. How to keep track of continuous query in a constantly changing environment [76].

Should queries be answered approximately [53] in case of querying the update intensive data such as location. Since the location information may be incomplete, new models of query answering that include data acquisition at run time are needed [35]. New methods for dynamic and distributed query optimization will have to be developed in order to handle different access costs from different locations. For example, same query may have different cost when formulated within local area wireless network or wide area environment. Therefore, the cost of query evaluation may depend on location of the querying site [34].

Massari et al. [54] present a query processing facility suitable for mobile database applications. The query model, called query by icons (QBI) considers the inherent limitations of mobile environment. It allows the construction of a database query with no special knowledge of how the database is structured and where it is located. The tools assist in the formation of the query during disconnections. A query is formulated in an incremental manner without accessing actual data in the remote database to materialize intermediate steps. Data are accessed and transmitted back to the mobile computer only when a complete query is materialized.

In [53], a query-processing model for mobile computing using summary databases (database stored in some predefined condensed form) is presented. The concept hierarchies are used to generate summary databases from the main database in various ways. It has been argued that in a mobile environment, it may be advantageous to relax one or other of the criteria to provide answers to queries that are both sound and complete with respect to the source data. This will enhance availability and would provide a more optimal use of data during periods of disconnection and to enable efficient utilization of low bandwidth and restricted memory size. The model for query processing proposed uses concept hierarchies and summary databases at run time to return approximate queries when access to the main database is either undesirable or unavailable. This model is able to provide varying levels of approximate answer to queries that occur at a mobile host using the summary database stored either locally at mobile host (MU) or remotely at BSs. They also discuss some cost-benefit analyses involving storage, transmission and query-processing costs.

Due to the limited bandwidth, direct answering of queries from a mobile device will likely jam the communication and thus will be very slow. Instead, it is better to feed the information progressively as the client refines the query. For examples, when a traveler drives to a place, she uses her PDA to find about interesting local attractions. To answer her query, the server can first send a list of attractions with brief description in text, and then send thumbnail images and directions for some of the attractions, and eventually full size images, maps, or video clips for the interested attractions. Such practices of informa-

tion filtering have been done in an ad hoc way by many Web sites. It is interesting to develop a theoretical data model to facilitate the process.

In the paper [26], we propose to extend the multi-layered database, and explore its potential and effectiveness in intelligent query answering in mobile environments. A multi-layered database **(MLDB)** is a database composed of several layers of information, with the lowest layer corresponding to the primitive information stored in a conventional database, and with higher layers storing more general information extracted from lower layers. Our model extends previous models by adding generalization operators to the model.

Other issues in query processing includes minimizing the search cost of locating data, preserving bandwidth, limiting the power consumption [3], controlling the precision of data to limit the computational and communication expenses.

6. Mobile transaction processing

A transaction in mobile environment is different than the transactions in the centralized or distributed databases in the following ways.
- The mobile transactions might have to split their computations into sets of operations, some of which execute on mobile host while others on stationary host. A mobile transaction shares their states and partial results with other transactions due to disconnection and mobility.
- The mobile transactions require 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 transaction, states of 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 disconnections.
- The mobile transactions should support and handle concurrency, recovery, disconnection and mutual consistency of the replicated data objects.

To support mobile transactions, 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 re-

duce 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.

Semantic based transaction processing models [16,65] have been extended for mobile computing in [79] to increase concurrency by exploiting commutative operations. These techniques require caching large portion of the database or maintain multiple copies of many data items. In [79], fragmentability of data objects have been used to facilitate semantic based transaction processing in mobile databases. The scheme fragments data objects. Each fragmented data object has to be cached independently and manipulated synchronously. That is, on request, a fragment of data object is dispatched to the MU. On completion of the transaction, the mobile hosts return the fragments to the BS. Fragments are then integrated in the object in any order and such objects are termed as reorderable objects. This scheme works only in the situations where the data objects can be fragmented like sets, stacks and queues.

In optimistic concurrency control based schemes [42], cached objects on mobile hosts can be updated without any co-ordination but the updates need to be propagated and validated at the DBS for the commitment of transactions. This scheme leads to aborts of mobile transactions unless the conflicts are rare. Since mobile transactions are expected to be long-lived due to disconnection and long network delays, the conflicts will be more in mobile computing environment.

In pessimistic schemes in which cached objects can be locked exclusively, mobile transactions can be committed locally. The pessimistic schemes lead to unnecessary transaction blocking since mobile hosts can not release any cached objects while it is disconnected. Existing caching methods attempt to cache the entire data objects or in some case the complete file. Caching of 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 MU allows only a small number of objects can be cached at any given time.

In [59], the concept of transaction proxies is introduced to support recovery. For each transaction submitted at an MU, a dual transaction called proxy is submitted to the base station. The proxy transaction includes the updates of the original transaction. Proxy transaction takes the periodic backup of the computation performed at mobile host.

Similar to above, in [67], the notion of twin-transaction was introduced which essentially replicate the process of executing transactions. In twin-transaction model, each write's request will be mirrored and two equivalent transactions will be created. In this way, if a mobile host is disconnected, the transaction execution can still be proceed. In [59], disconnection was not discussed.

Dynamic object clustering has been proposed in mobile computing in [56,57]. It assumes a fully distributed system, and the transaction model is designed to maintain the consistency of the database. The model uses weak-read, weak-write, strict-read and strict-write. The decomposition of operations is done based on the consistency requirement. Strict-read and strict-write have the same semantics as normal read and write operations invoked by transactions satisfying ACID properties. A weak-read returns the value of a locally cached object written by a strict-write or a weak-write. A weak-write operation only updates a locally cached object, which might become permanent on cluster merging if the weak-write does not conflict with any strict-read or strict-write operation. The weak transactions use local and global commits. The local commit is same as pre-commit of [48] and global commit is same as final commit of [48]. However, a weak transaction after local commit can abort and is compensated. In [48], a pre-committed transaction does not abort, hence require no undo or compensation. A weak transaction's updates are visible to other weak transactions whereas prewrites are visible to all transactions.

Lu and Satyanarayanan [45] present a new transaction model using isolation-only transactions (IOT). The model supports a variety of mechanisms for automatic conflict detection and resolution. IOTs are sequences of file accesses that unlike traditional transactions have only isolation property. Transaction execution is performed entirely on the client and no partial result is visible on the servers. IOTs do not provide failure atomicity, and only conditionally guarantee permanence. They are similar to weak transactions of [56].

An open nested transaction model has been proposed in [17] for modelling mobile transactions as a set of subtransactions. They introduce reporting and co-transactions. A reporting transaction can share its partial results, can execute concurrently and can commit independently. Co-transactions are like co-routines and are not executed concurrently. The model allows transactions to be executed on disconnection. It also supports unilateral commitment of subtransactions, compensating and non-compensatable transactions. The author claims that the model minimizes wired as well as wireless communication cost. However, not all the operations are compensated [17], and compensation is costly in mobile computing.

Transaction models for mobile computing that perform updates at mobile computers have been developed in [17,56]. These efforts propose a new correctness criterion [17] that is weaker than the serializability. They can cope more efficiently with the restrictions of mobile and wireless communications.

In [48-50], we look mobile transaction more as a concurrency control problem and provide database consistency. We incorporate a prewrite operation [47] before a write operation in a mobile transaction to improve data availability. A prewrite operation does not update the state of a data object but only makes visible the value that the data object will have after the commit of

the transaction. Once a transaction received all the values read and declares all the prewrites, it can pre-commit at mobile host (i.e., computer connected to unreliable communication) and the remaining transaction's execution is shifted to the stationary host (i.e., computer connected to the reliable fixed network). Writes on database, after pre-commit, take time and resources at stationary host and are therefore, delayed. This reduces network traffic congestion. A pre-committed transaction's prewrite values are made visible both at mobile and stationary hosts before the final commit of the transaction. This increases data availability during frequent disconnection common in mobile computing. Since the expensive part of the transaction's execution is shifted to the stationary host, it reduces the computing expenses (e.g., battery, low bandwidth, memory etc.) at mobile host. Since a pre-committed transaction does not abort, no undo recovery needs to be performed in our model. A mobile host can cache only prewrite values of the data objects, which will take less space, time, and energy and can be transmitted over low bandwidth.

A kangaroo transaction (KT) model was given in [25]. It incorporates the property that transactions in a mobile computing hop from a base station to another as the mobile unity moves. The mobility of the transaction model is captured by the use of split transaction [63]. A split transaction divides on going transactions into serializable subtransactions. Earlier created subtrans-action is committed and the second subtransaction continues its execution. The mobile transaction is split when a hop occurs. The model captures the data behavior of the mobile transaction using global and local transactions. The model also relies on compensating transaction in case a transaction aborts. The model in [48] has the option of either using nested transactions or split transactions. However, the save point or split point of a transaction is explicitly defined by the use of pre-commit. This feature of the model allows the split point to occur in any of the cell. Unlike KT model, the earlier subtransaction after pre-commit can still continue its execution with the new subtransaction since their commit orders in the model [48] are based on pre-commit point. Unlike KT, the model in [48] does not need any compensatory trans-action.

In [38], a distributed lock management scheme is presented. It allows a read unlock for an item to be executed at any copy site of that item; the site may be different from the copy site on which read lock is set. The proposed scheme utilizes the replicated copies of data items to reduce the message cost incurred by the mobility of the transaction host.

In a multidatabase environment with mobile computers involved, the nature of computing is such that the user may not wait for the submitted global transaction to complete before disconnected from the network. In [85], a basic architectural framework to support transaction management is multidatabase systems is proposed and discussed. A simple message and queuing facility is suggested which provides a common communication and data exchange pro-

tocol to effectively manage global transactions submitted by mobile workstations. The state of global transactions is modelled through the use of subqueues. The proposed strategy allows a mobile workstations to submit global transactions and then disconnected itself from the network to perform some other tasks thereby increasing processing parallelism and independence.

A transaction management model for the mobile multidatabase is presented in [21], called toggle transaction management technique. Here site transactions are allowed to commit independently and resources are released in timely manner. A toggle operation is used to minimized the ill-effects of the prolonged execution of long-lived transaction.

In most of the above papers, no comparative performance evaluation of models is presented. We observe that there is a need to investigate the properties of mobility, which can impact most the transaction processing. Also, there is a need to evaluate various transaction processing algorithms with respect to performance, response time, throughput and may be a new paradigm like quality of service (QoS) in the transaction management in the area such as e-commerce.

6.1. Broadcast disk and transaction processing

In traditional client-server systems, data are delivered from servers to clients on demand. This form of data is called pull-based. Another interesting trend is push-based delivery in a wireless environment. In wireless computing the stationary server machines are provided with a relative high bandwidth channel which supports broadcast delivery to all mobile clients located inside the cell. In a push-based delivery, server repetitively broadcast data to clients without specific request. Clients monitor the broadcast and retrieve data items they need as they arrive on the broadcast channel. This is very important for a wide range of applications that involve dissemination of information to a large number of clients. Such applications include stock quotes, mailing lists, electronic newsletters, etc. Broadcast in a mobile computing has number of difficulties. How to predict and decide about the relevance of the data to be broadcasted to clients. One way is that the clients may subscribe their interests to services [76]. The server also needs to decide about either sending the data periodically [1] or aperiodically. Mobile clients are also resource poor and the communication environment is asymmetric. The problem also is to maintain the consistency of broadcast data. The commercial system, which supports the concept of broadcast data delivery, has been proposed [77]. Recently, broadcast [18] has received considerable attention in the area of transaction processing in mobile computing environment.

Pitoura and Chrysanthis [60] addresses the problem of ensuring consistency and currency of client read-only transactions when the values are being broadcast in the presence of updates at the server. They broadcast additional

control information in the form of invalidation reports, multiple versions per item and serializability information. They propose different methods that vary in complexity and volume of control information transmitted and subsequently differs in response times, degree of concurrency, and space and processing overheads. The proposed methods are combined with caching to improve query latency.

Pitoura and Chrysanthis [61] exploit versions to increase concurrency of client read-only transactions in the presence of updates at the servers Invalidation reports are used to ensure the currency of reads. They broadcast older versions along with new values. The approach is scalable as it is independent of the number of clients. Performance results show that the overhead of maintaining older versions is low and at the same time concurrency increases. On the same line, [72] presents an approach for concurrency control in Broadcast environments. They propose a weaker notion of consistency, called update consistency, which still satisfy the mutual consistency of the data.

Barbara [9] reports transactions support in a mobile database environment with the use of broadcast facility. Mobile clients use broadcast data to verify if transactions are serializable. Lee Sang-keun et al. [40] present an optimistic concurrency control protocol with update timestamps to support transactional cache consistency in a wireless mobile computing environment using broadcast. They implement the consistency check on accessed data and the commitment protocol in a truly distributed fashion as a part of cache invalidation process with most burden of consistency check being downloaded to mobile clients. They achieve improved transaction throughput in comparison to [9] and it minimizes wireless communication required for supporting mobile transactions.

7. Mobile database research directions

We see the following as upcoming mobile database research directions.

7.1. Location-dependent query processing

We present ideas for processing queries that deal with location-dependent data [51]. Such queries we refer as location-dependent. Location can be a subject of more complex aggregate queries. For example, finding the number of hotels in the area you are passing or looking for a mobile doctor closest to your present location. Hence, the location information is a frequently changing piece of data. The objective is get the right data at different locations for processing a given query. The results returned in response to such queries should satisfy the location constraints with respect to the point of query origin,

where the results are received, etc. We propose to build additional capabilities into the existing database systems to handle location-dependent data and queries.

We present some examples to recognize the problems of accessing correct data when point of contact changes. Data may represent SSN of a person, or maiden name, or 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 SSN of a person remains the same no matter from which location it is accessed. This is not true in the case of sales tax data. The value of the sales tax depends on the place where sales 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 therefore, identify the type of data whose value depends on the set of criteria established by the location and another type of data, which is not subject to the constraints of a location. There is a third type of data that is sensitive to the point of query. We illustrate this data with the following example. Consider a commuter who is travelling in a taxi initiates a query on his laptop to find the nearby hotels in the area of its current location. The answer to this query depends on the location of the origin of the query. Since the commuter is moving he may receive the result 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 movement does not affect the answer to the query 'find the cheapest hotel'. The former depends on the location where as later on the object characteristics. Madria et al. [51] discuss the data organization issues in location-dependent query processing.

In [34], queries with location constraints are considered. They consider the query such as 'find the nearest hotel from my current position'. The main objective there is to minimize the communication cost to retrieve the necessary information to answer the query. The authors suggest greedy heuristics to solve the problem.

In MOST project [82,83], they consider a database that represents information about moving objects and their location. They argue that existing DBMS's are not well equipped to handle continuously changing data, such location of moving objects. They address the issue if location modelling by introducing the concept of dynamic attribute (whose value keeps changing), spatial and temporal query languages and indexing dynamic attributes.

7.2. View maintenance in mobile computing

Accessing on-line database from the mobile computer may be expensive due to limited uplink bandwidth, and also due to the fact that sending messages consume lot of energy which is limited in the portable battery. These two

problems can be solved by maintaining a materialized view i.e., storing the tuples of the view in the database at the mobile computer. This view will be updated as on-line database changes using wireless data messages. This will also localize access, thus improving access time. Therefore, to better deal with the problem of disconnection, reliability, and to improve response time, the view should be materialized at the mobile computer. The view maintenance will involve location-dependent data, time-dependent data, and dynamic allocation of a materialized view in the fixed and mobile network. The another problem is dynamic allocation of a materialized view in the mobile network [32]. Some more work in this directions have been reported in [22]. Another issue concerns the divergence of the materialized view at the mobile computer from the on-line database. In other words, how closely should the materialized view reflect the on-line database. An approach to divergence given in [33] involves parame-terizing each read at the mobile computer with the amount of divergence from the latest version it can accept. Another approach given in [73] allows the user to specify triggers on the on-line database, the view is updated when the trigger is satisfied. Recently, this problem has also been discussed in a position paper [69] to emphasis the importance of data warehousing view maintenance in mobile computing environment.

There is a need to develop view maintenance algorithms in case of data warehousing environment where relational data is broadcasted. Similarly, there is a need to do change management in the web data where changes are broadcast periodically and mobile host should capture the data and make the cached web data consistent.

7.3. Workflows in mobile environment

Workflow management systems are growing due to their ability to improve the efficiency of an organization by streamlining and automating business processes. Workflow systems have to be integrated with mobile computing environment [4] in order to co-ordinate disconnected computing to enhance the system's resilience to failures. Specific issues that arise here include how the workflow models can co-ordinate tasks that are performed when mobile users work in disconnected mode and when they cross wireless boundaries. Also location sensitive activities might have to schedule to use an organization's resources effectively. Current workflow systems do not seem to have any provision to handle these requirements.

Alonso et al. [4] discuss how disconnected workflow clients can be supported while preserving the correctness of the overall execution and allowing co-or-dinated interactions between the different users. The required activities (i.e., applications and data) are downloaded onto the mobile computer before performing a planned disconnection. The activities are performed in a dis-connected mode and the results are then uploaded after reconnection at which

time exceptions that occurred during the disconnected mode of operations are also handled. When mobile users cross the borders of wireless cells, hand-off (i.e., a mobile user's session is to migrate to a new information server) may need to be performed from the old workflow server to a new server. Consistency issues that arise when workflow instances migrate are to be handled carefully. For location sensitive routing of a mobile user's request, the modelling primitives should have provision to specify geographic information in the workflow definition.

7.4. Digital library services in mobile computing

Digital libraries bring about the integration, management, and communication of gigabytes of multimedia data in distributed environment. Digital library data includes texts, figures, photographs, sound, video, films, slides, etc. Digital library system currently envisions users as being static when they access information. It is expected in the near future that users will have access to a digital library through wireless access. Provision to access digital library services through wireless networks is required by a wide range of applications from personal to research to customize business computing. Providing digital library services to users whose location is constantly changing, whose network connections are through a wireless medium, and whose computational power is low necessitates modifications to existing digital library systems. The queries in digital library are complex and involve processing, navigation, searching, and presenting of distributed heterogeneous repositories of multimedia data.

7.5. Mobile web and e-commerce

There is a need to bring web on the mobile platform. Imagine a taxi that is equipped with mobile computer and the passenger there would like to browse web pages while waiting for its destination. The limited bandwidth will be a bottleneck in such a scenario. Another interesting application may be e-commerce on the mobile web. All these applications are ready to go on the disconnected, highly unreliable, limited bandwidth and unsecured platform where application demands reliability and security. Some efforts in this direction have appeared in [78], which is a WWW system designed to handle mobile users. It allows the documents to refer and react to current location of clients. In [75], they address the issue of mobile web browsing through a multi-resolution transmission paradigm. The multi-resolution scheme allows various organizational units of a web document to be transferred and browsed according to the information contents. This will allow better utilization of limited bandwidth. Similar effort has been reported in [29].

7.6. Mobile data security

Security is a prime concern in mobile databases due to nature of communication medium. New risks caused by mobility of users, portability of computers can compromise on confidentiality, integrity and availability including accountability. In a mobile database environment, it may be a good idea if data can be summarized [53,55,58,74], or only metadata can be stored on mobile platform and more detailed data can be kept on mobile service station (MSS) only. The higher frequency of disconnection also requires a more powerful recovery model. Such situations offer attackers the possibility of masquerade as either mobile host or MSS. This needs more robust authentication service [13]. Another issue is to maintain the privacy of location data of mobile hosts. Ideally only mobile user and home agent should have knowledge about mobile hosts current position and location data. All user identification information including message origin and destination has to be protected. In order to achieve anonymous communication, aliases can be used or communication can be channeled through a third trusted party. Further more, the identity of users may also need to be kept secret from other MSSs if required. Access based control policies can be adapted to provide data security on mobile platform.

8. Conclusions

Management of data in the mobile computing environment offers new challenging problems. Existing DBMS software need to be upgraded to adapt them to the new environment. To be able to do so, the critical parameters need to be understood and defined. Mobility brings in new dimension to the existing solutions to the problems in distributed databases. We have surveyed some of the problems and existing solutions in that direction. We have highlighted the merits and demerits of existing solutions. We have identified some of the upcoming research areas that require rethinking due to nature and constraints of mobile computing environment. The upcoming mobile database research directions discussed here will be the centres of attractions among mobile database researchers in years to come. There is a need to explore these issues further and improve the existing solutions offered in that direction.

References

[1] S. Acharya, R. Alonso, M. Franklin, S. Zdonik, Broadcast disks: data management for asymmetric communication environments, in: Proceedings of the ACM SIGMOD Conference, CA, 1995.

[2] R. Alonso, H.F. Korth, Database system issues in nomadic computing, in: Proceedings of ACM SIGMOD Conference on Management of Data, 1993, pp. 388-392.

[3] R. Alonso, S. Ganguly, Query optimisation for energy efficiency in mobile environments, in: Proceedings of the 1993 International Workshop on Foundations of Models and Languages for Data and Objects, Aigen, Austria, 1993.

[4] G. Alonso, R. Gunthor, M. Kamath, D. Agrawal, El. Abbadi, C. Mohan, Exotica/FMDC: handling disconnected clients in a workflow management systems, in: 3rd International Conference on Co-operative Information Systems, May, 1995.

[5] B. Awerbuch, D. Peleg, Online tracking of mobile users, Journal of ACM 42 (5) (1995) 1021-1058.

[6] V. Anantharaman, M.L. Honig, U. Madhow, V.K. Wei, Optimisation of a database hierarchy for mobility testing in a personal communication network, Performance Evaluation 20 (13) (1994).

[7] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[8] B. Daniel, Mobile computing and databases: a survey, IEEE Transactions on Data and Knowledge Engineering (1999).

[9] D. Barbara, Certification reports: supporting transactions in wireless systems, in: Proceedings of 17th International Conference on Distributed Computing Systems, 1997, pp. 466-473.

[10] B.R. Badrinath, A. Acharya, T. Imielinski, Structuring distributed algorithms for mobile hosts, in: 14th IEEE International Conference on Distributed Computing Systems, June, 1994, pp. 21-28.

[11] B. Bruegge, B. Benninington, Applications of mobile computing and communications, IEEE Personal Communications 3 (1) (1996).

[12] B.R. Badrinath, T. Imielinski, Replication and mobility, in: 2nd IEEE Workshop on the Management of Replicated Data, November, 1992, pp. 9-12.

[13] B. Bhargava, S.B. Kamisetty, S. Madria, Fault tolerant authentication in mobile computing, Special Session, New Paradigms in Computer Security, in: International Conference on Internet Computing (IC'2000), Las Vegas, USA, pp. 395-402.

[14] D. Barbara, T. Imielinski, Sleepers and workaholics: caching strategies in mobile environments, VLDB Journal (1995).

[15] B.R. Badrinath, T. Imielinski, A. Virmani, Locating strategies for personal communication networks, in: IEEE GLOBECOM workshop on Networking for Personal Communications Applications, December, 1992.

[16] N. Barghouti, G. Kaiser, Concurrency control in advanced database applications, ACM Computing Surveys 23 (3) (1991) 269-317.

[17] P.K. Chrysanthis, Transaction processing in a mobile computing environment, in: Proceedings of IEEE workshop on Advances in Parallel and Distributed Systems, October, 1993, pp. 77-82.

[18] Chung, B. Bhargava, S. Madria, Taxonomy of data management via broadcasting in a mobile computing environment, in: Mobile Computing: Implementing Pervasive Information and Communication Technologies, Kluwer Academic Publishers, Dordrecht, 2001.

[19] G. Cho, L.F. Marshall, An efficient location and routing scheme for mobile computing environments, IEEE Journal on Selected Areas in Communications 13 (5) (1995).

[20] J. Cai, K.L. Tan, B.C. Ooi, On incremental cache coherency schemes in mobile computing environment, in: Proceedings of IEEE International Conference on Data Engineering (ICDE), 1997.

[21] Dirckze, L. Gruenwald, A toggle transaction management technique for mobile multidatabases, in: ACM Proceedings of International Conference on Information and Knowledge Management (CIKM), 1998.

[22] G. Dong, M. Mohania, Algorithms for view maintenance in mobile databases, in: Proceedings of the First Australian Workshop on Mobile Computing and Databases and Applications (MCDA'96), Melbourne, Australia, 1996.

[23] S. Dolev, D.K. Pradhan, Modified tree structure for location management in mobile environments, Computer Communications 19 (1997) 335-345.

[24] M.R. Ebling, Evaluating and improving the effectiveness of caching for availability, Ph.D. Thesis, Department of Computer Science, Carnegie Mellon University, 1997.

[25] M.H. Eich, A. Helal, A mobile transaction model that captures both data and movement behaviour, ACM/Baltzer Journal on Special Topics on Mobile Networks and Applications (1997).

[26] Y. Fu, S. Madria, Multi-layered databases for intelligent query answering in mobile environments, in: International Workshop on Reliable and Secure Applications in Mobile Environments, New Orleans (also invited paper in NSF workshop), October, 2001.

[27] M. Faiz, A. Zaslavsky, Database replica management strategies in multidatabase systems with mobile hosts, in: 6th International Hong Kong Computer Society Database Workshop, 1995.

[28] D.J. Goodman, Trends in cellular and cordless communications, IEEE Communication Magazine (1991).

[29] M. Gaedke, M. Beigl, G. Hans-Werner, C. Segor, Web content delivery to heterogeneous mobile platforms, in: ER Workshops Proceedings as Lecture Notes in Computer Science, vol. 1552, Springer, Berlin, 1998.

[30] J. Gray, P. Helland, P. O'Neil, D. Shasha, The dangers of replication and a solution, in: Proceedings of ACM SIGMOD International Conference on Management of Data, 1996, pp. 173-182.

[31] J.S.M. Ho, F. Akyildiz, Dynamic hierarchical database architecture for location management in ocs networks, IEEE Transactions on Networking 5 (5) (1997).

[32] Y. Huang, P. Sistla, O. Wolfson, Data replication for mobile computers, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1994.

[33] Y. Huang, P. Sistla, O. Wolfson, Divergence caching in client-sever architectures, in: Proceedings of the Third International Conference on Parallel and Distributed Systems (PDIS), Austin, TX, September, 1994, pp. 131-139.

[34] T. Imielinksi, B.R. Badrinath, Wireless mobile computing: challenges in data management, Communications of ACM 37 (10) (1994).

[35] T. Imielinksi, B.R. Badrinath, Querying in highly distributed environments, in: Proceedings of the 18th VLDB, August, 1992, pp. 41-52.

[36] J. Ioannidis, G.Q. Maquire, The design and implementation of a mobile networking architecture, in: USENIX Winter 1993 Technical Conference, January, 1993.

[37] R. Jain, Reducing traffic impacts of PCS using hierarchical user location databases, in: Proceedings of the IEEE International Conference on Communications, 1996.

[38] J. Jing, O. Bukhres, A. Elmagarmid, Distributed lock management for mobile transactions, in: Proceedings of the 15th International Conference on Distributed Computing Systems, Vancouver, Canada, June, 1995.

[39] R. Jain, Y. Lin, C. Lo, S. Mohan, A caching strategy to reduce network impacts of (PCS), IEEE Journal on Selected Areas in Communications 12 (8) (1994).

[40] Lee Sang-keun, Hwang Chong-Sun, Yu HeonChang, Supporting transactional cache consistency in mobile database systems, in: ACM International Workshop on Data Engineering for Wireless and Mobile Data Access, 1999.

[41] M. Krishnamurthi, M. Azizoglu, A.K. Somani, Optimal location management algorithms for mobile networks, in: Proceedings of the Fourth ACM International Conference on Mobile Computing and Networking (MOBICOM'98), 1998, pp. 223-232.

[42] J. Kisler, M. Satyanarayanan, Disconnected operation in the coda file system, ACM Transactions on Computer Systems 10 (1) (1992).

[43] P. Krishna, N.H. Vaidya, D.K. Pradhan, Static and dynamic location management in mobile wireless networks, Journal of Computer Communications 19 (4) (1996) (special issue on Mobile Computing).

[44] Y. Lin, Reducing location update cost in PCS network, IEEE Transactions on Networking 5 (1) (1997) 25-33.

[45] Q. Lu, M. Satyanaraynan, Improving data consistency in mobile computing using isolation-only transactions, in: Proceedings of the Fifth Workshop on Hot Topics in Operating Systems, Orcas Island, Washington, May, 1995.

[46] S.K. Madria, Transaction models for mobile computing, in: Proceedings of 6th IEEE International Conference on Network (ICON'98), World Scientific, Singapore, July, 1998.

[47] S.K. Madria, B. Bhargava, System defined prewrites to increase concurrency in databases, in: Proceedings of First East-European Symposium on Advances in Databases and Information Systems (in co-operation with ACM-SIGMOD), St.-Petersburg, September, 1997.

[48] S.K. Madria, B. Bhargava, Improving availability in mobile computing using prewrite operations, Distributed and Parallel Database Journal (2001).

[49] S.K. Madria, B. Bhargava, A transaction model for mobile computing, in: Proceedings 2nd IEEE International Database and Engineering Application Symposium (IDEAS'98), Cardiff, UK, 1998.

[50] S.K. Madria, B. Bhargava, On the correctness of a transaction model for mobile computing, in: 9th International Conference on Database and Expert System Applications (DEXA'98), Vienna, Austria, Lecture Notes in Computer Science, vol. 1460, Springer, Berlin, 1998.

[51] S.K. Madria, B. Bhargava, E. Pitoura, V. Kumar, Data organization issues in location dependent query processing in mobile computing environment, in: Proceedings of 4th East-European Symposium on Advances in Databases and Information Systems (in co-operation with ACM-SIGMOD), Prague, Czech Republic, 2000.

[52] S. Mohan, R. Jain, Two user location strategies for PCS, IEEE Personal Communications Magazine 1 ((1), 1st quarter) (1994).

[53] S.K. Madria, M. Mohania, J. Roddick, A query processing model for mobile computing using concept hierarchies and summary databases, in: Proceedings of the 5th International Conference on Foundation for Data Organization (FODO'98), Japan, November, 1998 (Also, appeared as book chapter in K. Tanaka, S. Ghandeharizadeh (Eds.), Information Organization and Databases, Kluwer Academic Publishers, Dordrecht, 2000).

[54] A. Massari, S. Weissman, P.K. Chrysanthis, Supporting mobile database access through query by icons, Distributed and Parallel Databases: An International Journal (1996).

[55] E. Pitoura, Transaction-based co-ordination of software agents, in: 9th International Conference on Database and Expert System Applications (DEXA'98), Vienna, Austria, Lecture Notes in Computer Science, vol. 1460, Springer, Berlin, August, 1998.

[56] E. Pitoura, B. Bhargava, Building Information Systems for Mobile Environments, in: Proceedings of 3rd International Conference on Information and Knowledge Management, 1994, pp. 371-378.

[57] E. Pitoura, B. Bhargava, Maintaining consistency of data in mobile computing environments, in: Proceedings of 15th International Conference on Distributed Computing Systems, June, 1995 (Extended version to appear in IEEE Transactions on Knowledge and Data Engineering, 1999).

[58] E. Pitoura, B. Bhargava, Dealing with mobility: issues and research challenges, Technical Report TR-93-070, Department of Computer Sciences, Purdue University, 1993.

[59] E. Pitoura, B. Bhargava, Revising transaction concepts for mobile computing, in: Proceedings of the 1st IEEE Workshop on Mobile Computing Systems and Applications, December, 1994, pp. 164-168.

[60] E. Pitoura, P.K. Chrysanthis, Scalable processing of read-only transactions in broadcast push, in: Proceedings of IEEE International Conference on Distributed Computing Systems, 1999.

[61] E. Pitoura, P.K. Chrysanthis, Exploiting versions for handling updates in broadcast disks, in: Proceedings of VLDB, 1999.

[62] E. Pitoura, I. Fudos, An efficient hierarchical scheme for locating highly mobile users, in: ACM Proceedings for International Conference on Information and Knowledge Management (CIKM), 1998.

[63] C. Pu, G. Kaiser, Hutchinson, Split-transactions for open-ended activities, in: Proceedings of the 14th VLDB Conference, 1988.

[64] E. Pitoura, G. Samaras, Locating objects in mobile computing, TKDE 13 (4) (2001) 57l—592.

[65] K. Ramamritham, P.K. Chrysanthis, A taxonomy of correctness criterion in database applications, Journal of Very Large Databases 4 (1) (1996).

[66] K. Ravindran, K. Shah, Casual broadcasting and consistency of distributed data, in: 14th International Conference on Distributed Computing Systems, June, 1994, pp. 40-47.

[67] A. Rasheed, A. Zaslavsky, Ensuring database availability in dynamically changing mobile computing environment, in: Proceedings of the 7th Australian Database Conference, Melbourne, Australia, 1996.

[68] M. Satyanarayanan, Mobile information access, IEEE Personal Communications 3 (1) (1996).

[69] I. Stanoi, D. Agarwal, A. El Abbadi, S.H. Phatak, B.R. Badrinath, Data warehousing alternatives for mobile environments, in: Proceedings of ACM International Workshop on Data Engineering for Wireless and Mobile Access, Seattle, Washington, 1999.

[70] M. van Steen, F.J. Hauck, P. Homburg, A.S. Tanenbaum, Locating objects in wide-area systems, IEEE Communication Magazine (1998) 2-7.

[71] M. Satyanarayanan, J.J. Kistler, B. Kumar, et al., Coda: a highly available file system for a distributed workstation environment, IEEE Transaction on Computers 39 (4) (1990).

[72] J. Shanmugasundaram, A. Nithrakashyap, R. Sivasankaran, K. Ramamritham, Efficient concurrency control for broadcast environments, in: Proceedings of the ACM SIGMOD Conference, 1999.

[73] N. Shivakumar, J. Widom, User profile replication for faster location lookup in mobile environments, in: Proceedings of the 1st ACM International Conference on Mobile Computing and Networking (MOBICOM'95), October, 1995, pp. 161-169.

[74] P. Sistla, O. Wolfson, Temporal conditions and integrity constraints in active database systems, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, May, 1995, pp. 269-280.

[75] M.T. Stanley, Y. Hong Va Leong, D. McLeod, A. Si, On multi-resolution document transmission in mobile web, SIGMOD Record 28 (3) (1999) 37-42.

[76] D.B. Terry, D. Goldberg, D.A. Nichols, B.M. Oki, Continuous queries over append-only databases, in: Proceedings of the ACM-SIGMOD International Conference on Management of Data, June, 1992.

[77] Vitria Technology Inc. Available from www.vitira.com.

[78] G.M. Voelker, B.N.B. Mobisaic, An information management system for mobile wireless computing environment, in: T. Imienlinski, H. Korth (Eds.), Mobile Computing, Kluwer Academic Publishers, 1996, pp. 375-395.

[79] G.D. Walborn, P.K. Chrysanthis, Supporting semantics-based transaction processing in mobile database applications, in: Proceedings of 14th IEEE Symposium on Reliable Distributed Systems, September, 1995, pp. 31-40.

[80] Wu Shiow-yang, Y. Change, An active replication scheme for mobile data management, in: IEEE Proceedings of 6th DASFAA, Taiwan, 1999.

[81] O. Wolfson, S. Jajodia, Distributed algorithms for dynamic replication of data, in: Proceedings of the Symposium on Principles of Database Systems, CA, 1992, pp. 149-163.

[82] O. Wolfson, A.P. Sistla, C. Sam, Y. Yelena, Updating and querying databases that track mobile units, Distributed and Parallel Databases 7 (3) (1999) 257-387.

[83] O. Wolfson, X. Bo, C. Sam, L. Jiang, Moving objects databases: issues and solutions, in: Proceedings of SSDBM, 1998, pp. 111-122.

[84] K. Wu, P.S. Yu, M. Chen, Energy Efficient caching for wireless mobile computing, in: Proceedings of the 12th International Conference on Data Engineering, New Orleans, February, 1996.

[85] L.H. Yeo, A. Zaslavsky, Submission of transactions from mobile workstations in a co-operative multidatabase processing environment, in: Proceedings of the 14th IEEE International Conference on Distributed Computing Systems (ICDCS'94), June, 1994.