# Database System Issues in Nomadic Computing

Rafael Alonso
Henry F. Korth
Matsushita Information Technology Laboratory
Princeton, NJ 08540

## Abstract

Mobile computers and wireless networks are emerging technologies that will soon be available to a wide variety of computer users. Unlike earlier generations of laptop computers, the new generation of mobile computers can be an integrated part of a distributed computing environment, one in which users change physical location frequently. The result is a new computing paradigm, *nomadic computing*. This paradigm will affect the design of much of our current systems software, including that of database systems.

This paper discusses in some detail the impact of nomadic computing on a number of traditional database system concepts. In particular, we point out how the reliance on short-lived batteries changes the cost assumptions underlying query processing. In these systems, power consumption competes with resource utilization in the definition of cost metrics. We also discuss how the likelihood of temporary disconnection forces consideration of alternative transaction processing protocols. The limited screen space of mobile computers along with the advent of pen-based computing provides new opportunities and new constraints on database interfaces and languages. Lastly, we believe that the movement of computers and data among networks potentially belonging to distinct, autonomous organizations creates serious security problems.

## 1 Introduction

At present, "mobile computing" consists of users carrying laptop computers, typically IBM-PC-compatible machines, running system software originally designed for non-mobile environments. These machines connect to other computers via a modem and a telephone line when the user is traveling, and by direct network connection when the user is in the office. The office computers typically contain the primary copies of data, with needed data downloaded to the laptop computer before a trip, and uploaded upon return. The user has full responsibility for ensuring that the right files are uploaded and downloaded.

Technical advances in recent years have made laptops larger in capacity and speed, yet smaller in physical size. Such advances, when viewed from a systems software point of view, have been evolutionary rather than revolutionary. However, the state of the art is now reaching a point where we are seeing a radical shift in the way laptop computers are used. Laptops have become capacious enough to hold databases that would have been called "very large" not too long ago, and fast enough to support complex database operations. Furthermore, wireless network technology permits these computers to interact as integral parts of a distributed system. Finally, the set of potential users for these computers has been enlarged by the introduction of pen-based interfaces and ever-smaller physical packaging (including "palmtop computers").

The increased capabilities of these computers is such that users have begun to use these systems in much the same manner as they use their desktop workstations. But at this point, both the volume of data that a mobile computer can hold and the degree of potential connectivity with other computers is such that users cannot reasonably be expected to manage their data by manually uploading and downloading files. Instead, the mobile computer must operate as part of a distributed system – but subject to the special constraints imposed by the mobile computer hardware itself, wireless network technology, and the fact that the computer and its user are physically moving. The term *nomadic computing* has been used to describe this computing paradigm.

This paper looks at nomadic computing from a database system perspective. We contend that nomadic computing changes fundamentally some of the assumptions underlying existing protocols, models, and algorithms for database systems. This requires that several database research areas be revisited within this context.

The rest of this paper is organized as follows. In Section 2, we present some technical details about


SIGMOD /5/93/Washington, DC,USA

the state of the art in mobile computers and wireless networks. Section 3, which contains the main technical contribution of this paper, discusses the impact of nomadic computer on several aspects of database system internals. Section 4 remarks on the problem of designing a database language and user interface suitable for a pen-based computer with a small-size screen. Concluding remarks are in Section 5.

## 2 Mobile and Wireless Technology

There is currently on the market a wide array of devices that are popularly considered to be mobile computers [Nad92]. We will argue that some, such as the HP 95 (with a packet radio attachment) or pen-centric data entry devices (such as that carried by United Parcel Service personnel) are too limited in power to be considered true general-purpose mobile computers. The typical mobile system we have in mind is any of a number of PC-compatible machines, Intel 386SX/486SX-based, with 4 megabytes of main memory and more than 100 megabytes of disk storage. If current technological trends continue, the CPU speed, memory size, and disk capacity of these computers will roughly double every eighteen months. However, a number of other limitations will remain in the foreseeable future.

Currently, mobile systems have a relatively limited battery capacity, usually two to three hours of capacity under "normal" use (see the performance tests in [O'M92]). This figure is not expected to increase by more than 30% over the next five years [Eag91]; the current limitations on power per weight will continue with a small improvement in the near future. There is ongoing work on low-power chips (e.g., AT&T's Hobbit processor), as well as research on lower-voltage architectures [CSB92] (halving the voltage swing required for a system results in power savings of a factor of four); however, we expect that users will demand fast processors on their mobile machines (to be able to run the same applications that they run on their workstations with comparable performance), resulting in an increased demand for power.

The screen size of mobile systems will most likely not increase very much in the near future (it may even decrease!), given that these machines' small size is one of their main virtues. Barring a major technological breakthrough (e.g., virtual reality eyeglasses, or foldable screen technologies), we do not expect current limitations to be alleviated.

There are (roughly speaking) two types of networking mechanisms for mobile computers. The first consist of local area networks such as NCR WaveLAN (essentially a "wireless Ethernet") which provide connectivity within a small geographical area (e.g., within a building), at relatively low cost in terms of power and money. The other alternative is the use of cellular or packet radio modems (e.g., Ericsson GE's Mobidem) which offer connectivity over much larger geographical zones, but at correspondingly increased costs. Typical bandwidths for local area solutions are 2 megabits/sec (WaveLAN), and 8 Kbits/second for the long-haul radio ones (Mobidem packet radio). While these numbers will improve over the next few years, it is safe to assume that network bandwidth will remain a major performance bottleneck for system design in the near future.

Finally, the reliability of the overall distributed computing environment remains a question mark. It is not presently known how reliable network links will be in practice, especially when users start taking these systems to places with large background electrical noise. It is not easy to predict how many times a dropped computer will result in a communication gap (not to mention more catastrophic failures). It is clear however, that these systems will be used under much harsher conditions than the typical workstation. Thus, we feel that the frequency and duration of network partitions will be at least an order of magnitude larger for these systems than is currently common in non-mobile distributed systems. Although the above points suggest that the mobile environment is more failure-prone than the traditional one, certain of these "failures" are foreseeable. A user may be able to pre-announce 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. The result is that a key failure mode of a mobile computing environment is "frequent, foreseeable disconnection." The system can take action to "recover" from such failures before they occur, as we shall see in the next section.

## 3 Database System Internals

### 3.1 Query Processing

It is not unusual for changes in the computing environment to result in modifications to the query-processing component of a database system. For example, consider the evolution from strictly centralized database systems to the early distributed systems based upon relatively low bandwidth wide-area networks. That change forced query optimizers to focus mostly on network transmission costs in selecting optimal query plans, which led to the study of semi-join algorithms [BC81]. As local area networks became commonplace, communication costs became less important, and optimizers went back to considering a variety of resource costs. Many of the characteristics of nomadic computing force us once again to reconsider old choices. In this section we concentrate on exploring the impact of two particular ones: the networking environment, and the reliance of mobile systems on battery power.

The probable networking environment for mobile database systems will be a highly heterogeneous one, with a wide mix of bandwidths and architectures [IB92]. As can be expected, bandwidth specifications will motivate changes to the optimizer algorithms. But there will also be new problems to consider. For example, the dollar cost of sending information over a wireless network are expected to be high. (The monthly charge in US dollars for RAM Mobile Data's Mobitex is between $50 and $80, with a 1000 character message costing about $0.25 [Hay92].) This cost may lead some designers of query optimizers to focus solely on minimizing the financial expenses of transactions. As a second example, consider the use of a cellular modem to communicate. The added dollar cost of communication start versus continuing the phone call will provide an incentive for considering query processing strategies for long-lived transactions that do not rely on frequent short communications but instead require fewer (albeit longer) conversations.

As we pointed out in Section 2, current battery technology seems to place a limit of two hours or so on user sessions. Since heavy database system users will probably suffer even shorter battery lives, one of the problems that one must consider is how to limit the impact of database use upon this resource. One possibility is to develop new query optimization algorithms that will select plans based on their energy consumption. Thus, the usual criterion of maximizing the number of transactions per second may be replaced by one of minimizing power per transaction. We direct the interested reader to [AG92] for some initial work in this area discussing a variety of new query optimization algorithms for nomadic database systems.

The constraints of the nomadic computing environment will make approximate answers to queries more acceptable than in traditional database systems, since the alternative to an approximate answer may be no answer at all. The results potentially adaptable to the nomadic environment include quasi-copies [ABGM90].

Another aspect of query processing for nomadic computing relates to the nature of queries themselves, as noted in [IB92]. For example, parameters to a query may be expressed relative to the current location of the mobile computer from which the query is entered. They may depend also on the direction and speed with which the the computer is moving. Or perhaps location-dependent data may change during the time the query is being processed.

These aspects of nomadic queries affect not only optimization, but also query languages and user interfaces.

### 3.2 Transaction Processing

Applications of nomadic computing will demand various transactional and transaction-like services. These range from such data-processing tasks as remote order entry, to more complex, interactive tasks such as calendar management, CAD-like activities, and saga-like [GMS87] activities. Thus, the transaction model for nomadic computing will need to include aspects of long-transaction models and sagas. Because mobile computers have a nontrivial amount of local storage, transaction processing is at least as complex as in a traditional distributed database system. Since users need to be able to work effectively during periods of disconnection, mobile computers will require a substantial degree of autonomy in transaction management. This introduces aspects of the multidatabase problem.

Thus, the nomadic computing paradigm includes features of many of the research areas currently of interest to the transaction-processing research community. However, as we shall see below, nomadic computing adds its own special variants to these issues.

#### 3.2.1 Failures

Frequent, 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 processes 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 in support of 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.
- The mobile computer may take action to "declare itself down" by removing itself from quorums for distributed protocols in which it may be participating. This may allow the distributed protocol to handle the disconnection with less overhead than in current models in which the disconnection is discovered only after it occurs.

Another aspect of fault tolerance unique to the mobile computing environment is the instability of stable storage. In practice, disk storage is treated as acceptably stable for log records in non-mobile systems. Highly reliable systems use replicated logs. However, a mobile computer is uniquely vulnerable to a total catastrophic failure due to the user dropping the machine, data destruction by an airport security system, or even the loss or theft of the entire machine. Therefore, the remark above about transfer of log records from the mobile computer to a non-mobile one may indeed be a critical activity prior to a foreseen disconnection.

When a mobile computer reconnects, there is likely to be a pent-up demand for communication. Local transactions on the mobile machine may need remote data to complete; remote transactions may need local data; and the system may have critical housekeeping functions to complete (such as the transfer of log records from the mobile computer). Since the reconnection may be of only limited duration, these communications tasks need to be ordered by priority based on system integrity issues, response time to user requests, and efficient resource utilization.

### 3.2.2 Distributed Computing Issues

Although the nomadic computing environment is *just* a distributed system, the cost/performance aspects of the system differ from those of traditional distributed systems.

Communication to/from a mobile computer will be more costly than among non-mobile computers. This cost is not just network dollar costs, but also cost in terms of the battery of the mobile computer. Because receiving consume less power than broadcasting, the communication cost function is not symmetric. Batching of data to be sent to/from a mobile computer will lower overall cost.

These changes in the communication cost structure, as well as the failure-mode considerations discussed above will alter the relative practicality of known distributed protocols, and may lead to new protocols as well. We conjecture that primary-copy algorithms will serve well for data used primarily by a particular mobile computer.

### 3.3 Security

Many of the security issues in nomadic computing are not specific to database systems. Others are database specific. We begin by addressing briefly security issues not specific to database applications and then consider some that pertain primarily to databases.

Since mobile computers appear and disappear on various networks, prevention of impersonation of one machine by another is problematic. When a mobile computer is taken away from its local environment, data it sends and receives are subject to theft and to unauthorized copying. A network that allows visiting mobile computers to connect to it cannot perform the type of packet filtering now used as a security mechanism, since certain foreign packets will be legitimate packets destined for the visiting machine.

A mobile machine visiting a "foreign" environment will consume some resources of that environment. These include network bandwidth at a minimum. For database applications, the power considerations previously discussed suggest the use of disk storage and perhaps CPU cycles from the foreign environment as well. Because the demands placed on the environment by a database application may be nontrivial, accounting (in much the same sense as traditional time-sharing systems) may be needed for both billing purposes and for limiting the impact of visiting machines on the performance of the foreign environment.

Perhaps the most interesting aspect of database security in the nomadic computing environment is that of trust between system components. Traditionally, database systems based their operation and their security on that of the underlying operating system. Clearly, some such degree of trust is needed for resource sharing between a mobile computer and a foreign environment to work successfully. However, it is not clear what the exact nature of this degree of trust is. Ideally, trust should be limited to standard systems software components that can authenticate themselves to the mobile computer.

The above remarks focus on the concerns the user of the mobile machine may have about security. However, the administrator of the foreign environment has security concerns as well. These concerns are much greater than the current mode of "mobile computing" in which a user in a foreign environment is logged in to a local guest account from which the user telnets to his/her home environment. In the nomadic computing paradigm, a guest machine may harm its host – either accidentally or maliciously. The possibility of such harm is much greater than that likely by the typical user of a guest account. Can bounds be placed on the damage done, while allowing the mobile machine the access to local resources that it needs?

## 4 Human-Computer Interface

Traditional database query languages, especially SQL, cannot easily be used on pen-based computers. The state of the art in handwriting recognition is inadequate for input as lengthy as the typical SQL query. Thus, a graphical user interface in which queries are expressed by pointing to elements of the schema or to data items themselves are needed. Earlier work on graphical user interfaces (see, e.g., [Kim86]) mix use of a keyboard with use of a mouse. They also exploit the large screens of typical workstations in presenting the user interface. For database access in nomadic-computing mode, the interface must require an absolute minimum of text entry in order to support keyboardless, pen-based computers. Furthermore, the interface cannot be dependent on the availability of a large screen due to the small physical size of mobile computers.

These two constraints are being examined as part of the Neon project at MITL. In order to reduce the amount of screen space required to represent a database schema, we are using concepts from the nested relational model to represent parts of the schema by a single attribute. A universal-relation-based approach is combined with nested relations to allow queries to be specified more simply. Ambiguities that result from the use of a universal relation are resolved by pen-based selection of alternatives.

Although our current prototype still requires a keyboard, we are reducing our reliance on the keyboard with the goal of eventually being totally pen-reliant. Among the features helping to reduce the reliance on the keyboard are pen-based selection of attributes from the schema, pen-based selection of data values from the database for insertion into a query, and pen-based selection of data from the domain of an attribute for insertion into a query. The latter two items, in particular, present serious performance demands on the system as well as on the human factors of the interface. We are continuing to address those issues.

We are designing our system with an expected heterogeneous database environment in mind. This includes not only the ability to access multiple relational databases, but also the ability to access other types of database systems and even non-database information sources such as file systems (this requires exploring such seemingly strange ideas as joining a relation with a file). Preliminary work is described in [AHK92].

## 5 Conclusion

Many of the challenges that have been met by the database research community over the years involved the adaptation of formal database concepts to the practical constraints of storage devices, networks, processing speed, memory size, etc. Although these constraints change constantly, these changes are usually evolutionary. Only infrequently are there revolutionary shifts that require database researchers to reconsider old approaches in light of new technology. It is our view that nomadic computing is one of these revolutionary shifts.

At the current time, systems research pertaining to nomadic computing has been mostly in the areas of networks and operating systems. However, the issues introduced go beyond those areas and directly affect database systems. Nomadic computing introduces new concerns regarding power management, stable data storage, wireless communication, and user interface. We have noted above how these issues affect query processing, transaction processing, distributed computing, security, and database query languages. The research community is only beginning to address these issues. The pace at which new hardware is being introduced to the mobile computer market suggests that a window of opportunity has opened for the database research community to have significant practical impact.

## References


[ABGM90] R. Alonso, D. Barbará, and H. Garcia-Molina. Data caching issues in an information system. *ACM Transactions on Database Systems*, 15(3):359–384, September 1990.

[AG92] R. Alonso and S. Ganguly. Energy-efficient query optimization. MITL-TR-33-92, Matsushita Information Technology Laboratory, Princeton, NJ, 1992.

[AHK92] R. Alonso, E. Haber, and H. F. Korth. A database interface for mobile computers. In *Proceedings of the 1992 Globecom Workshop on Networking of Personal Communication Applications*, December 1992.

[BC81] P. A. Bernstein and D.-M. W. Chiu. Using semi-joins to solve relational queries. *Journal of the ACM*, 28(1):25–40, January 1981.

[CSB92] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen. Low-power CMOS digital design. *IEEE Journal of Solid-State Circuits*, 27(4), April 1992.

[Eag91] Eager. Advances in rechargeable batteries pace portable computer growth. In *Proceedings of the Silicon Valley Personal Computer Conf*, 1991.

[GMS87] H. Garcia-Molina and K. Salem. Sagas. In *Proceedings of ACM-SIGMOD 1987 International Conference on Management of Data, San Francisco*, pages 249–259, 1987.

[Hay92] D. Hayden. The new age of wireless. *Mobile Office*, May 1992.

[IB92] T. Imielinski and B. R. Badrinath. Querying in highly mobile distributed environments. In *Proceedings of the Eighteenth International Conference on Very Large Databases, Vancouver*, pages 41–52, 1992.

[Kim86] H.-J. Kim. Graphical interfaces for database systems: A survey. In *Proceedings of the 1986 Mountain Regional ACM Conference, Santa Fe, NM*, 1986.

[Nad92] M. Nadeau. Portable computing - notebooks coming of age. *Byte (Outlook '92 Special Issue)*, January 1992.

[O'M92] C. O'Malley. Comparison report: The SL germination. *Mobile Office*, May 1992.