View Maintenance in Mobile Computing *

Ouri Wolfson †, Prasad Sistla† , Son Dao ‡, Kailash Narayanan† , Ramya Raj†

1 Introduction

Mobile users of portable computers will soon have online access to a large number of databases via wireless networks. For example, while on the road, passengers will access airline and other carriers schedules, and weather information; investors will access prices of financial instruments, sales people will access inventory and other corporate data, commuters will access traffic information. However, access to an online database may be unavailable when the user is out-of-range of the wireless network. Furthermore, sending access-requests from the mobile computer to the online database may be expensive due to limited uplink bandwidth, and also due to the fact that sending messages is a significant drain on the portable battery. These two problems can be alleviated by maintaining a materialized view at the mobile computer. This view will be updated using wireless data messages, as the online database changes. This will also localize access, thus improving access time to the view.

Therefore, to better deal with the problem of disconnection and to improve response time, the view should be materialized at the mobile computer. As explained above, sometimes communication cost can also be reduced by materialized views; but not always. For example, if the materialized view at the mobile computer is accessed very seldom, whereas the view is updated very frequently, then the cost of wireless messages that update the view may become excessive. (For example, currently RAM Mobile Data Corp. charges on average \$0.08 per data message, the exact charge depending on the message size). In this case, assuming that cost minimization is the major consideration, the view better stay virtual (i.e. nonmaterialized) at the mobile computer; the view will be requested from the fixed network, when accessed at the mobile computer. In other words, the optimal *allocation scheme*, i.e. the set of mobile or stationary computers at which the view is materialized, may change dynamically as the access pattern to the view changes throughout the network. The optimal allocation scheme also depends on the cost structure in the fixed and wireless network. Thus, we are also conducting research into *dynamic views*, i.e. views that are materialized according to different allocation schemes at different points in time.

In conclusion, the objective of this paper is to outline the major issues related to maintaining personalized views in a network of fixed and mobile computers. Although the issue of view maintenance has been studied extensively in the literature (see [1] for surveys), our research concentrates on new problems introduced by distribution and mobility. New problems are introduced by the fact that the connection between the materialized view at the mobile computer and the online database may vary widely in terms of cost, reliability, and capacity. Specifically, wide-area wireless networks

*This research was supported by NSF Grant No. IRI-9408750

†Electrical Engineering and Computer Science Department, University of Illinois Chicago, Illinois 60607

‡Hughes Research Laboratories, Information Sciences Laboratory, Malibu, CA

are costly, unreliable, and have a small bandwidth, whereas local-area wireless networks suffer much less from these limitations. We are currently building a software system for view maintenance in mobile computing that addresses these problems.

The rest of this paper is organized as follows. In section 2 we outline the mobile computing problems in view maintenance, and some possible solutions. In section 3 we describe the high-level software architecture for view maintenance in a mobile environment. In section 4 we describe the prototype system that we are currently building in order to test our approach.

2 Mobile Computing Issues in View Maintenance

In this section we discuss communication problems in view maintenance, and our approach to the solution.

The first problem concerns location-dependent and time-dependent, continuously changing views. An example of a location-dependent view is: retrieve the set of motels (with availability and prices) within a radius of 5 miles. This view changes as the mobile computer moves. A view that is both, time and location dependent is the weather and traffic within 1 mile of the mobile computer, in the direction of travel of the mobile computer. Such views are continuously being updated as the mobile computer moves, and the question is how often the view should be recomputed, and how frequently it should be refreshed at the mobile computer. A possible solution to this problem would provide the mobile computer with some default function according to which the view changes; updates will occur only when the view change does not correspond to the function ([11]). For example, the function will indicate no traffic problems, and an update will occur only when such a problem is identified.

The second problem addressed by our research is *dynamic views*, i.e. dynamic allocation of a materialized view in the fixed and mobile network. Initial results in this area have been presented in [2] for the wireless network, and in [9,10] for the fixed network. We are combining and generalizing the previous results to deal with various communication cost models, and various network topologies. We are also considering access cost in addition to communication cost. Specifically, reading some online databases may involve an access cost in addition to communication cost (similarly, calling a 900 telephone number involves an access cost). This additional cost is taken into consideration when determining the optimal allocation scheme. Additionally, there may be a dichotomy between the read/write granularity, and the allocation granularity. For example, read/writes may be at the granularity of tuples, whereas data-allocation may have to take place at the relation granularity. Thus, we are taking this dichotomy into consideration in managing dynamic views.

The third communication issue addressed by our research concerns various transmission strategies for maintaining the materialized/virtual views. The three obvious transmission methods are point-to-point, broadcasting, and multicasting. In a cellular architecture, within a cell, broadcasting may be the only available method. However, communication to users residing in multiple cells may be possible in any of the three methods. The optimal method depends on the relative cost of the various methods, and on the number (and possibly location) of the recipients of an update to a materialized view. Related issues involve "batching" of updates for transmission purposes, and combining update methods. One possible combination was analyzed in [8], where invalidation requests were broadcasted, and view updates were requested and provided in a point-to-point mode. Other combinations are possible. One we are particularly interested in is dynamically switching between broadcasting and point-to-point transmission of view updates.

The fourth issue concerns the divergence of the materialized view at the mobile computer from the online database. In other words, how closely should the materialized view reflect the online database. On one extreme are protocols that do not provide any guarantee that limits the divergence between the two. On the other extreme are protocols that ensure that the view at the mobile computer is a replica of the online database. Here, each transaction that updates a related data-item at the online database updates the mobile computer view as well, and each update of the view at the mobile computer is propagated atomically to the online database. Maintaining complete consistency between the online database and the view may slow down transaction processing significantly, since concurrency control and commitment have to be performed across a slow and unreliable wireless network. An approach to divergence that we discussed in [7] involves parameterizing each read at the mobile computer with the amount of divergence from the latest version that it can accept. Another approach allows the user to specify triggers on the online database; when a trigger is satisfied, the view is updated (see [6]). Some other approaches were discussed in [3,4,5]. In our research we are mainly concerned with providing the user with a set of parameters by which s/he can control how far the view is allowed to diverge from the online database, and the mechanisms for enforcing this limit.

The fifth issue concerns disconnection. This issue can be divided into two cases. The first case is when the materialized view at the mobile computer and the online database need to be updated atomically, i.e., in one transaction, and the second is when the materialized view and online database may be inconsistent. We are concerned with defining a set of parameters by which the user provides default options in case of disconnection. Such options will tell the system, for example, what to do with updates that are received by the online database while the mobile computer is disconnected. Should they be discarded or late-delivered when connection is re-established, and if late-delivered, in what form and order? The default options can be overridden when the disconnection is planned and voluntary (e.g. the user turns off her mobile computer). In other words, the user provides a set of options to be used when the disconnection is unexpected, and these options can be modified for a particular planned disconnection. In contrast, most existing work on disconnection involves methods of resolving inconsistencies that arise as a result of disconnection.

The last problem is transparent (to the user) view maintenance in the face of movement between wireless networks. For example, the user may move outside the range of a wireless LAN having one Megabits per second bandwidth, but stay within the range of a wireless WAN that has a bandwidth of 19 Kilobits per second. Assuming that the communication between the online database and the mobile computer can take place on either media, we would like to provide view maintenance over the LAN when in range, and transparently and automatically switch to the WAN when out of the LAN range. We would also like to switch automatically between wide-area networks, when necessary or beneficial, e.g. when a lower-cost network comes in range. Furthermore, we are studying how to build a stand-alone communication subsystem that provides dynamic and automatic switching between networks for general applications, not necessarily view maintenance. This involves, for example, determining a set of parameters by which the user tells the system when and how to select one of the available wireless networks.

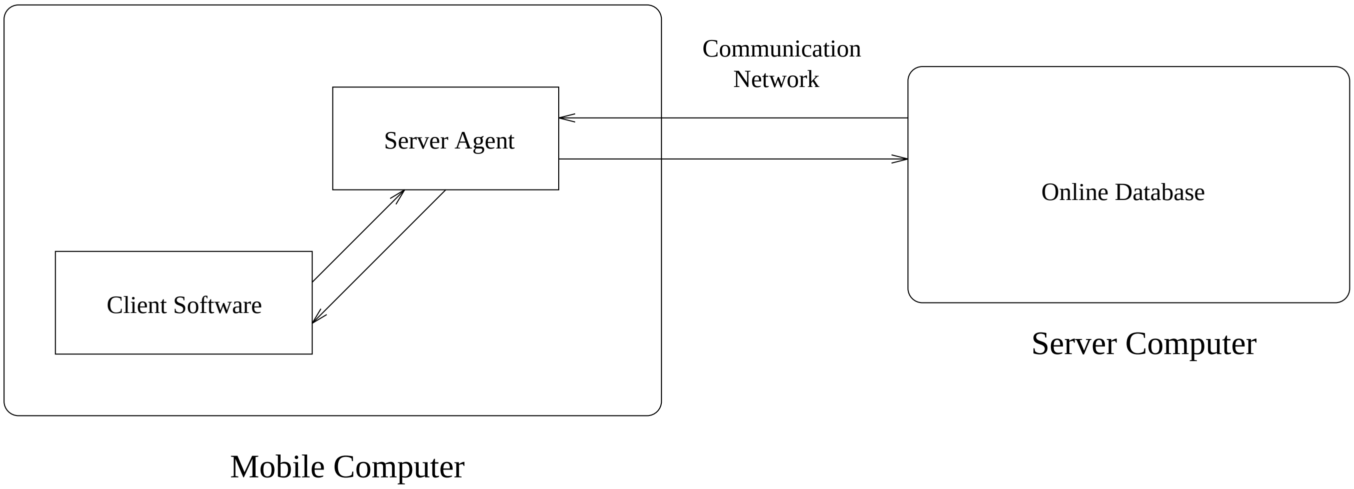

Figure 1: Software Architecture

3 Software Architecture

The environment consists of three independent functional entities, the database publisher, the network provider, and the customers. Each customer pays access charges to the database publisher, and the database publisher pays the network provider for the communication. Alternatively, the customer may pay communication charges directly to the network provider. In some cases the access charge is zero. For example, if the user is a salesperson that accesses the inventory information in the corporate database, access charge will probably be zero.

In order to reduce access and communication cost, the user can materialize at the mobile computer a view that is divergent from the one at the database publisher. In other words, the mobile computer will not receive every update to the materialized view, but only selective ones.

In the general case, each mobile computer can communicate with the fixed network via two separate channels. One is a broadcast channel, and the other is a point-to-point channel.

In order to access the database, a user has to install in his/her mobile computer a module called the Server Agent (SA), provided by the database publisher. The software architecture is illustrated in figure 1.

In the rest of this section we outline the services provided by the SA to the user, and the interface between the SA and the fixed network.

The SA provides the following services for each view, x. These services are provided to the user, or to a software module operating in her behalf.

- *subscription*: Asynchronously, provide each update of x.
- *divergent-subscription*: As above, except that the SA is requested to maintain a view that can be divergent from the fixed-network database by a certain limit. The limit is specified using a language we are developing. Alternatively, the SA is provided with a set of conditions on the database. When such a condition is satisfied, the materialized view is refreshed.
- *subscription cancellation*: The user indicates to the SA not to provide further updates to the view x.
- *read* : Provide the current contents of x. This

request may be asynchronous in the sense that the user does not wait for x.

- *invalidation*: The SA indicates that there was an update of the x, without providing the new contents.
- *statistics*: Provide various statistics on the view update rate, or number of users, etc. For example, the user may request the number of view updates since the last read request s/he issued. This request can be combined with a read request, such that the view is refreshed and statistics are provided in one transmission.
- *cost structure*: Provide information about access and communication charges involved in reading/writing x.

Using these primitive services the user can implement higher level algorithms for dynamic view maintenance. For example, the user can implement the sliding window algorithm discussed in [2] in which the determination whether or not to materialize a view at mobile computer is performed based on a sliding window of the latest k requests; if the cost of the reads in the window is higher, then the view is materialized, otherwise it is not.

Similar services are provided by the fixed network to the Server Agent. The divergent-subscription capability requires that the database publisher supports triggers which indicate when to refresh the materialized view at the mobile computer. Thus, the active database research becomes relevant to maintenance of materialized views.

4 The Wireless View System

We are currently building a software system called Wireless-View, for implementing view maintenance in mobile computing. The system addresses the problems outlined in section 2, and it consists of a subsystem residing on the mobile computer, and another subsystem residing on the fixed network. Among other things, the system can be used for optimizing access and communication cost. These costs are optimized by selecting an appropriate dynamic view maintenance algorithm. Some possible algorithms are Subscription, On-demand, and the Sliding Window algorithm mentioned above. The appropriate algorithm depends on the access pattern to the view, and the access and communication cost structures, and the allowed divergence. These are provided as parameters to the Wireless-View system, which in turn selects a dynamic view maintenance algorithm. The algorithm is implemented using the services provided by the Server Agent.

The Wireless-View system currently runs on an Intel 486 laptop computer running MS Windows, with the server being a SUN Sparc station. We have experimented with Wavelan as a LAN, and with Radiomail as a WAN. The database consists of a relation of 20 stock prices that are continuously updated according to the stock market activity during an hour of trading.

The Wireless-View system will be further evaluated in a collaborative effort between Hughes Research Laboratories' WINS (Wireless Information Network System) laboratory, and the Distributed Computing and Databases laboratory at the University of Illinois at Chicago. With this collaboration, we will be able to extend our wireless network testbed to include Direct Broadcast Satellite (DBS) transmission using Hughes Network System (HNS) DirecPC product. DirecPC uses a modem for the uplink transmission, and it uses the VSAT satellite for the downlink transmission (bandwidth for the latter is 12MBS).

References

[1] J. Widom, Ed. Special Issue on Materialized Views and data Warehousing, *Data Engineering Bulletin, IEEE Computer Society* , June 1995, Vol 18, No.2.

[2] Y. Huang, P. Sistla, O. Wolfson, Data Replication for Mobile Computers, *Proceedings of the ACM Sigmod 1994, International Conference on Management of Data*, Mineapolis, MN, May 1994, pp. 13-24.

[3] E. Pituora and B. Bhargava, Maintaining Consistency of Data in Mobile Distributed Environments, Proc. of the 15th International Conf. on Distributed Computing Systems, Vancouver BC, May 1995.

[4] E. Pituora and B. Bhargava, Revising Transaction Concepts for Mobile Computing, Proc. of the IEEE Workshop on Mobile systems and Applications, Santa Cruz, CA Dec. 94.

[5] R. Alonso and H. Korth, Database System Issues in Nomadic Computing, *Proceedings of the ACM Sigmod 1993, International Conference on Management of Data*, Washington DC, May 1993.

[6] P. Sistla, O. Wolfson, Temporal Conditions and Integrity Constraints in Active Database Systems, *Proceedings of the ACM Sigmod 1995, International Conference on Management of Data*, San Jose, CA, May 1995, pp.269-280.

[7] Y. Huang, R. Sloan, O. Wolfson, Divergence Caching in Client-Server Architectures", *Proceedings of the third International Conference on Parallel and Distributed Information Systems (PDIS),* Austin, TX, Sept. 1994, pp. 131-139.

[8] D. Barbara, T. Imielinski, Sleepers and Workaholics: Caching Strategies in Mobile Environments, *Proceedings of the ACM Sigmod 1994, International Conference on Management of Data*, Mineapolis, MN, May 1994.

[9] O. Wolfson, S. Jajodia, Y. Huang, An Adaptive Data Replication Algorithm, submitted for publication.

[10] O. Wolfson and S. Jajodia, Distributed Algorithms for Adaptive Replication of Data. *Proceedings of the 11th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, San Diego, CA, June 1992, pp. 149-163.

[11] Sam Chamberlain, personal communication.