Editor: Ronald D. Williams, University of Virgina, Thornton Hall/Electrical Engineering, Charlottesville, VA 22903; phone, (804) 924-7960; fax, (804) 924-8818; Internet, rdw@virginia.edu

Mobile computing

M. Satyanarayanan, Carnegie Mellon University

In his book *Mind Children*, my colleague Hans Moravec draws an analogy between the seminal role of mobility in the evolution of biological species and its similar impact upon the development of computing systems.[1] Although Hans' comments are directed at robotic systems, his observation applies equally well to a much broader class of distributed computing systems involving mobile elements. Mobility will influence the evolution of distributed systems in ways that we can only dimly perceive at present.

The recent proliferation of portable computers, in conjunction with nascent high- and low-bandwidth cordless networking technology, will soon provide a pervasive hardware base for mobile computing. The stage is set for portable computing devices of every variety imaginable, from the handheld to the wearable. Telecommuting will be truly ubiquitous.

This exciting possibility brings with it new technical challenges. Mobile computing systems are constrained in important ways relative to static systems. Moreover, these constraints are intrinsic to mobility and are not just the shortcomings of current technology:

- *Mobile elements are resource-poor relative to static elements.* Regardless of future technological advances, a mobile unit's weight, power, size, and ergonomics will always render it less computationally capable than its static counterpart. While mobile elements will undoubtedly improve in absolute ability, they will always be at a relative disadvantage.
- *Mobile elements are more prone to loss, destruction, and theft than static elements.* A Wall Street stockbroker is more likely to be mugged on the streets of Manhattan and have his or her laptop stolen than to suffer the loss of a workstation in a locked office through theft or tampering. Even if security isn't a problem, portable computers are more vulnerable to loss or damage.
- *Mobile elements must operate under a much broader range of networking conditions.* A desktop workstation can typically rely on local or wide area connectivity (LAN/WAN). A laptop in a hotel room has only modem or ISDN (Integrated Services Digital Network) connectivity. Outdoors, a laptop with a cellular modem may experience only intermittent contact with its nearest cell.

These constraints violate many of the assumptions upon which today's distributed systems are based. Further, the future ubiquity of portable computers will result in mobile computing systems that are much larger than the distributed systems of today. Scalability will thus be a continuing concern. Past experience has shown that scale has to be treated as a primary influence rather than an afterthought in the design of distributed systems.[2]

A key requirement of mobile computing systems will be the ability to access critical data regardless of location. Data from shared file systems and databases must be made available to programs running on mobile computers. For example, a technician servicing a jet engine needs access to the engineering details of that engine as well as to its past repair records; similarly, a businessman who is continuing his work on the train home needs access to his business records. For an even more complex and potentially life-threatening problem, imagine an emergency medical team responding to a case of poisoning; it would need rapid access to a medical database describing poison symptoms and antidotes, as well as to the patient's medical records to determine drug sensitivity.

The need to access shared data implies interdependence between the elements of a mobile computing system. At the same time, the need for robustness when encountering network and remote site failures requires clients to be as autonomous as possible. By its very nature, then, mobility exacerbates the tension between autonomy and interdependence so characteristic of distributed computing.

In principal, mobility should be completely transparent to users. Transparency eliminates the need to be constantly aware of one's computing environment, thus allowing the user to focus on the real tasks at hand. Adaptation to a changing computing environment should be initiated by the system rather than by the user. Although perfect transparency is unattainable, that should not deter us from striving to come as close as possible to that ideal.

The Coda file system, built by my research group at Carnegie Mellon University, represents an initial step in providing such transparency.[3] Coda facilitates the use of shared data in mobile computers by simplifying precaching of files, allowing autonomous operation while disconnected, and transparently reintegrating changes upon reconnection. Work is under way to extend the system to exploit weak connectivity (low-bandwidth, intermittent, or both) when available. Since disconnected operation allows Coda to operate at zero bandwidth, we view weak connectivity as an opportunity rather than a problem. Our approach to exploiting weak connectivity is to use it for trickle-charging or discharging a mobile client's cache.

The design of Coda extends the file access paradigm of the Andrew file system to mobile environments.[4] Coda completely hides mobility, so an application cannot tell whether it is connected or disconnected from servers. Coda has the further advantage of being binary compatible with applications written for the Berkeley Software Distribution (BSD) Unix interface. Thus, for an important class of common applications, Coda achieves full transparency.

But the strategy of insulating applications from the limitations inherent in mobility can be pushed only so far. Supporting more ambitious applications will require making mobility visible in a controlled manner so that applications can modify their behavior to cope with the vagaries of a mobile communication environment.

Consider, for example, an application capable of displaying stored full-motion video images. Below a certain bandwidth and network quality, it will not be possible for the application to display the images in full-motion color. Although extensive compression will help, it cannot solve the problem completely. But if the application were also capable of displaying the image in slow-scan black and white, it could automatically do so when the bandwidth fell below a critical threshold. With this strategy, sophisticated applications would be able to sense and react to mobility in an application-specific manner, thereby minimizing user involvement. This in turn will allow the mobile client to offer the user the best service attainable at the current physical location.

For another, perhaps more broadly relevant, example, consider the resolution of conflicting updates made to shared data. Such conflicts may arise if an optimistic replication strategy is used to improve data availability in failure-prone mobile networks. Since resolution is an application-specific concept, masking such conflicts from users will require the system to invoke application-specific code to perform the resolution.[5] The net effect of this strategy is to decouple the detection of conflicts from their resolution.

Since different applications adapt differently to mobility, extensibility will be an important requirement for effective support of applications in mobile computing environments. Techniques that allow operating system functions such as caching and network transmission to be customized on an application-specific basis thus become extremely important. The challenge is to design and implement such extensibility without compromising efficiency.

References

1. H. Moravec, *Mind Children*, Harvard University Press, Cambridge, Mass., 1988.
2. M. Satyanarayanan, "The Influence of Scale on Distributed File System Design," *IEEE Trans. Software Engineering*, Vol. 18, No. 1, Jan. 1992, pp. 1-8.
3. J.J. Kistler and M. Satyanarayanan, "Disconnected Operation in the Coda File System," *ACM Trans. Computer Systems*, Vol. 10, No. 1, Feb. 1992, pp. 3-25; and M. Satyanarayanan et al., "Coda: A Highly Available File System for a Distributed Workstation Environment," *IEEE Trans. Computers*, Vol. 39, No. 4, Apr. 1990, pp. 447-459.
4. M. Satyanarayanan, "Scalable, Secure, and Highly Available Distributed File Access," *Computer*, Vol. 23, No. 5, May 1990, pp. 9-21.
5. P. Kumar and M. Satyanarayanan, "Supporting Application-Specific Resolution in an Optimistically Replicated File System," *Proc. Fourth IEEE Workshop on Workstation Operating Systems*, forthcoming from IEEE CS Press, Order No. 4000, Los Alamitos, Calif., 1993.

M. Satyanarayanan is an associate professor of computer science at Carnegie Mellon University. His address is School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3890; e-mail, satya@cs.cmu.edu.