Guest Editors’ Introduction

Experiences with Distributed Systems

Stephen F. Lundstrom, Stanford University

Duncan H. Lawrie, University of Illinois, Urbana-Champaign

The dramatic reductions in the cost of individual processors and memory has made a significant contribution to the increased use of distributed computing systems. Prototype systems are in place in research laboratories and universities, while a wide range of production systems are in place in business and industry.

Distributed computing systems have a wide range of configurations, but they generally consist of processor/memory pairs working together over a communications network, driven by software distributed across the processors. The distributed software that implements an operating system or a database system is the key to presenting a logically integrated system to its users, who do not need to be aware of the physical distribution of their system.

Despite these accomplishments, to date there is no set of consistent, widely used, theoretically based principles to guide the development of distributed computing systems. However, the lack of a complete theoretical basis for distributed computing systems need not inhibit the development of useful systems. Even without such a basis, many technical advances have been made by individuals, who then share them with others, who in turn accept useful concepts and add further innovations.

One channel for such communication is the annual Distributed Computing Systems Conference. This special issue of IEEE Software is based on the May 1984 conference held in San Francisco. The papers selected for this issue report actual implementation experience over a broad area of distributed computing systems.

The first article, “XMS: A Rendezvous-Based Distributed Software System Architecture,” by Neil D. Gammage and Liam M. Casey, describes a successfully operating distributed software development environment at BNR. This system is a good example of how a small number of basic concepts can be used to implement a powerful set of uniform, system-wide applications development capabilities.

The second article, “Helix: The Architecture of the XMS Distributed File System,” by Marek Fridrich and William Older, reports on the development of an important application within the XMS environment: the integration of the functional considerations of file system development and the implementation considerations of the distributed system environment.

Very few application programs have been developed specifically for a distributed environment. Usually, an application program in a distributed environment evolves from a program in a centralized environment that has
been patched to handle aspects of distribution. The third article, "Amaze: A Multiplayer Computer Game," by Eric J. Berglund and David R. Cheriton, describes an application developed especially for a distributed computing system. It was developed in an intentionally restricted communication environment, one limited to point-to-point messaging within the distributed system. The paper focuses on implementation techniques that can substantially reduce the overhead of communicating system state information. These techniques are important to distributed programs that maintain replicated data across the system, such as process control and robot control systems.

One of the most important aspects of distributed computing systems is the communications facilities between the system's various computer/memory pairs. The simplest capabilities are the point-to-point facilities used in the Amaze game. Distributed computing systems may also provide more sophisticated communications, where one source may send the same message to all other processing sites through a local area network. Many local area networks in use today are broadcast buses. The fourth article, "High-Level Broadcast Communication for Local Area Networks," by Thomas J. LeBlanc and Robert P. Cook, presents a high-level language implementation that directly supports broadcast communications. The article describes the broadcast capabilities supported by the language and discusses the performance speedup in distributed systems using broadcast communications.

The fifth article, "Multicast Communication on Network Computers," by Ariel J. Frank, Larry D. Wittie, and Arthur J. Bernstein, takes a more general view of communications. It explores group multicast techniques in distributed systems, defines eight criteria used to evaluate the suitability of known multicast techniques for group multicast, and discusses three techniques that are particularly suitable for groups on networked computers. The authors also describe the Micros project, a testbed for group organization and group multicast research.

Development of effective distributed system testbeds is critical to distributed computing system research and experimentation. These testbeds provide the integrated facility that researchers and developers of distributed computing systems use to examine design issues and to evaluate potential and alternative solutions to their problems. The sixth article, "The ARC Network: A Case Study," by Mark C. Paulk, describes the design and implementation of a network used for dynamic real-time distributed system simulation at the Ballistic Missile Defense Advanced Research Center. This article provides valuable insight into testbed development issues, an important issue since many systems do not have a realistic operational environment for validation testing.

We would like to acknowledge the excellent technical contributions at the 1984 Distributed Computing Systems Conference, from which these papers were selected, as well as the work of the authors in updating and improving their papers and the work of the independent reviewers who contributed helpful advice in structuring this special issue. We are looking forward to the 1985 conference in Denver this month.

Stephen F. Lundstrom is an associate professor of electrical engineering (research) at Stanford University. His research is centered on understanding how to project the performance of complete applications on highly concurrent and distributed computing systems. Lundstrom's other research interests include parallel processing, computer support of systems engineering, and tools to make technical communication more effective. He is a member of the IEEE and the ACM.

Duncan H. Lawrie is a professor of computer science, electrical and computer engineering, and supercomputing research and development at the University of Illinois, Urbana-Champaign. Lawrie is also a principal architect of the Cedar large-scale multiprocessor at the University of Illinois. His interests lie in high-performance computer architecture, especially communications schemes for tightly coupled microprocessors.

Lawrie is a fellow of the IEEE and a member of the ACM. He served as general chairperson of the Fourth International Conference on Distributed Computing Systems and as chairperson for conferences on the IEEE Computer Society Conferences and Tutorials Board.