Introduction — Distributed Computing

DURING the past years there has been a strong movement toward distributed computing systems. Driven by technological advances as well as user needs, the field of distributed computing is a natural evolution in computing history. This movement has caused computer system designers to reevaluate their views on system architecture, communication protocols, operating systems, programming languages, databases, and algorithms that have been used to design centralized computer systems. In response to this movement, the IEEE Computer Society established in 1979 a new technical committee on distributed processing and has sponsored an annual international conference on distributed computing systems since then. Recently, the Computer Society Governing Board has approved a new magazine, IEEE DISTRIBUTED COMPUTING, for publication beginning in the first quarter of 1987.

This special issue includes papers selected from the Fifth International Conference on Distributed Computing Systems held in Denver, CO, in May 1985, as well as papers in response to an open call for papers. Each of the 65 submitted papers was sent to four referees for review. Due to the space limitation, however, only 12 papers have been selected for inclusion in this special issue; other good papers, after revision and further review, may be published in future regular issues.

It is fitting that this issue begins with the paper by Gaudiot *et al.* on a distributed VLSI architecture for efficient signal and data processing. The goal of this architecture is to develop a high-performance multiprocessor system that is programmed in a high-level language as would be a single computer, while the multiprocessor organization is transparent to the user. In addition to enabling fast mapping of the application programs for various configurations, this approach allows easy fault recovery and can meet changing real-time performance requirements.

Among the many issues involved in multiprocessor design, development of a cost-effective interconnection network is essential. In their paper, Shin and Liu propose a cost-effective design of circuit-switching multistage interconnection networks that can increase the network bandwidth and reduce the network size through the concept of network overlapping and memory interleaving, instead of increasing the number of switches or adding buffers. Detailed analysis of the drastic reduction in hardware costs and the impact on system performance are also given.

Although communication protocols play an important role in computer networking and distributed computing, protocol design remains a difficult task. There is still much work to be done in the field of protocol specification, validation, verification, and implementation. Due to the complex hierarchical structure of protocols, the biggest problem in protocol design seems to be the lack of an easy, concise, and precise model for protocol's definition and manipulation. In his paper, Hoffman presents a methodology for formal specification of communication protocols. Based on a modified version of traces originally developed by Parnas and Bartussek for software specification, he first describes the trace language and then uses it to specify an ANSI standard protocol ADCCP as well as the DoD Internet Protocol.

Distributed operating systems (DOS's) must provide certain basic services such as communication, protection, resource management, reliability, and process (computation) abstraction. Most of the existing DOS designs tend to focus on only a few of these services. In their paper, LeBlanc and Friedberg describe the design of Heirarchical Process Composition (HCP), an object-oriented model of interprocess relationships that address all of these basic services. The major novelties of HCP lie in the extension of the process abstraction of processes and the provision of a rich set of structuring mechanisms for distributed computation. It is worth mentioning here that this paper won the best paper award at the Fifth International Conference on Distributed Computing Systems.

With recent advances in software, hardware, and communication technology for distributed computing, it may be possible to deal with distributed real-time systems in a more flexible way than in the past. Most current research on scheduling tasks with real-time constraints is restricted to multiprocessing systems and the scheduling of such tasks is done statically. In their paper, Stankovic *et al.* propose a task scheduling algorithm for loosely coupled distributed systems with the goal of achieving flexibility through the dynamic scheduling of tasks in a distributed and adaptive manner. Simulation studies are also used to evaluate the performance of their algorithm.

A distributed computing system may possess a large number of resources across the network. A resource can be logical such as a shared database, or can be physical such as a special-purpose functional unit. A task may be migrated because the local processor does not have the required resources or has failed. A task may also be executed remotely if the expected turnaround time is better. Resource scheduling maps requests to a pool of resources so as to optimize a combination of resource usage, response time, network congestion, and scheduling overhead. In their paper, Wah and Juang present a study of resource scheduling in an Ethernet-type local network. They propose an efficient method of utilizing the primitive operations of collision detection and broadcast for optimal resource scheduling in such an environment.

Replication of data in a distributed database system serves many purposes. In order to increase availability and enhance reliability, it is desirable to support the replication of critical and frequently used data at multiple sites. However, this will introduce a problem called mutual consistency, i.e., how to keep all the copies of data identical. In their paper, Sarin *et al.* present an approach to distributed database design that emphasizes high availability in the face of network partitions and other communication failures. Mutual consistency of replicated copies of data is ensured by using timestamps to establish a known total ordering on all updates issued, and by a mechanism that ensures the same final result regardless of the order in which a site actually receives these updates.

Local-area networks (LAN's) offer high-speed communication between distributed system components, such as intelligent workstations and shared file and printer servers. Most distributed computing systems designed for LAN's have adopted connectionless (also called datagram) protocols at the logical-link control (LLC) layer, because of datagram's simplicity and low overhead. However, if packet losses are frequent or if the LAN's are connected to long-haul networks like the ARPANET, connection-oriented (also called virtual-circuit) protocols may be more appropriate at the LLC layer. In their paper, Meister *et al.* investigate the impact of using a connection-oriented LLC protocol for file transfer in a LAN environment. Their results show that the performance penalty of a connection-oriented LLC protocol, in terms of the throughput between the file servers, can be substantial, but can be significantly reduced if an acknowledgment-accumulation strategy is used.

The four papers selected for inclusion in the correspondence section are all of high quality. Zwanenepoel describes an implementation of Unix pipes in the *V*-System and presents measurements of its performance. Ellis considers the problem of distributing data structures and develops a distributed version of an extensible hash file useful in a distributed database. Livny and Manber propose an extension of the token-passing ring protocol to allow a special type of arithmetic to be performed directly on the node's interface. Finally, Burton presents a simple functional-language feature that can be used to control speculative work on parallel machines.

The Guest Editor would like to thank the authors of the 12 papers cited above and those who submitted their papers for consideration in this issue. Special thanks are due to the 136 referees who did an outstanding job in reviewing the 65 papers within the very tight schedule required for this special issue. Last but not least, I should like to express my sincere appreciation to Tse-yun Feng, Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTERS, for giving me the opportunity to guest edit this special issue and for his constant encouragement and support.

MING T. (MIKE) LIU, *Guest Editor*
Dep. Comput. Inform. Sci.
The Ohio State Univ.
Columbus, OH 43210-1277

Ming T. (Mike) Liu (M'65–SM'82–F'83) received the M.S.E.E. and Ph.D. degrees from the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, in 1961 and 1964, respectively.

He stayed on the faculty of the Moore School as Assistant Professor of Electrical Engineering until 1969, when he joined The Ohio State University, Columbus, where he is presently Professor of Computer and Information Science. Since 1974 he has been actively involved in research and development of local networking and distributed computing, and published over 80 technical papers in this and related areas.

Dr. Liu was a Past Chairman of the IEEE Computer Society Technical Committee on Distributed Processing and served as Program Chairman of the Fifth International Conference on Distributed Computing Systems. Currently, he is a member of the Computer Society Governing Board, Chairman of the Tutorials Committee, and Chairman of the Eckert–Mauchly Award Committee. Since 1982 he has been an Editor of the IEEE TRANSACTIONS ON COMPUTERS. He is also a member of the Association for Computing Machinery and Sigma Xi.