Guest Editors’ Introduction
Performance Evaluation of Multiple Processor Systems

GENERAL REMARKS

Performance evaluation means to study, to analyze, and to optimize the flow of data and control information in computer systems. In doing so, the main objectives are to formally describe the real behavior of computers, to define and determine characteristic performance measures, and to develop tools for the design of optimal hardware structures and system software.

Performance influences the design, development, procurement, operation, management, and use of computer systems. Interest in performance evaluation techniques, including measurements, simulation, analytic techniques, and approximate solution algorithms, has increased in the past decade. During the same decade, technological advances have deeply changed the systems under study, requiring new performance evaluation techniques. In particular, the trend towards multiple processor systems, covering the spectrum from the tightly coupled multiprocessor organizations to geographically distributed processors, poses many new problems in evaluating their performance.

Consequently, in this Special Issue, we sought to draw together papers that either deepen our understanding of multiple processor system performance or advance the state of the art in measurement, analytic modeling, and simulation.

DISCUSSION OF THIS ISSUE

In studying the performance of computer systems, various techniques are available to model system behavior and workload. A central decision in modeling is choosing the level of detail, which lies between two extremes: either we may meticulously describe and analyze the real behavior or choose a more abstract and simplified model. The first case allows a detailed analysis of hardware and system software bottlenecks, while the second alternative may provide some first-order estimates of the overall throughput.

Limited memory bandwidth has often been a primary restriction in the design of tightly coupled multiprocessor systems with many CPU's. Memory bandwidth may be increased through cache memories. Two papers in this issue address the problems with the use of shared and private caches. The paper by Yeh et al. presents a shared-cache organization for a parallel-pipelined multiprocessor system with very few access conflicts. They use an analytic model to derive performance bounds and compare them to simulation results. Briggs and Dubois study the use of a private cache between a processor and the shared memory. An approximate model is used to obtain the processor utilization and the speedup improvements.

Marsan et al. present three modeling techniques frequently used to describe the behavior of a bus-structured multiprocessor system: Stochastic Petri nets, finite state Markov chain models, and queueing network models. Their advantages and drawbacks are discussed. Performance estimates from each technique are compared against each other and against results of both detailed simulation as well as measurements from a real triple processor system with synthetic load.

A designer of a multiple computer system faces many tradeoff issues. Improving our understanding of the impact of design decisions on system performance requires proper instrumentation and subsequent experimentation. The instrumentation issue is addressed by three papers in this issue. Segall et al. present an integrated instrumentation environment for multiprocessors, and give examples for the use of their approach for Cm* under both the StarOS and Medusa operating systems. Their work should provide a useful framework for the design of the instrumentation of multiple processor systems.

The paper by Fromm et al. presents the instrumentation approach taken for the Erlangen General Purpose Array. They present their measurement results, and then discuss how they used the measurements for validating the analytic model of their system.

Classical single processor system performance characteristics vary strongly with application and problem size. These difficulties are even more pronounced in highly parallel systems, such as the 4096 processor DAP system. Parkinson and Liddell demonstrate these effects using standard performance measures such as operation times, MIPS, FLOPS, GLOPS, speedup, and efficiency. The importance of choosing algorithms for parallel computation in such a way as to make best uses of the parallelism of the hardware is illustrated by examples.

In the classical modeling technique, one usually assumes concurrent processes to be independent of each other; on the other hand, it is also standard to assume that processes which are dependent on each other, e.g., I/O and CPU processes, take a sequential turn. Little research considers I/O and CPU overlap and recognizes that, nowadays, programs are decomposed into well-defined cooperating subtasks and are processed concurrently.

Heidelberger and Trivedi attack these problems for a specific program behavior: jobs are split into two or more concurrent tasks and are executed in parallel, possibly sharing various system resources. The parent task resumes execution only upon the completion of all its spawned tasks. An analytic, closed queueing network model is presented, and two ap-
proximate solutions are proposed: a hierarchical decomposition method and an iterative technique.

The cost–performance issue frequently dominates a design. Reed and Schwetman present cost–performance bounds for some multiple computer systems. They derive asymptotic bounds based on bottleneck analysis of queueing networks. The results are used to examine the performance/cost function in a network as the number of processors increases.

A major point of interest in designing multiple processor systems is reliability analysis. Mitrani and King summarize in their correspondence interesting results on the basic $M/M/N$ queueing system. Jobs processed on different processors are, therefore, independent of each other. The tradeoff between performance and reliability is studied under the assumption of two different patterns of breakdown and repair.

**Summary**

We believe that the information contained in this Special Issue—theoretical contributions as well as experiences with real systems—will be of lasting value to workers in the field of multiple processor systems, for system designers as well as performance evaluation specialists. The importance of performance analysis is recognized, and many helpful tools are available. But as computer systems and application complexity grow, new problems arise and challenge us.

We would like to acknowledge the help of many colleagues in soliciting and reviewing papers. Their dedicated efforts and numerous suggestions to the authors enhanced the papers. We also would like to thank the Associate Editor, Dr. S. Lavenberg, for suggesting this Special Issue and for his valuable aid in its preparation.

**ASHOK K. AGRAWALA, Guest Editor**

Univ. Maryland
College Park, MD 20742

**ULRICH HERZOG, Guest Editor**

Friedrich Alexander Univ. Erlangen-Nürnberg
Erlangen, West Germany

---

Ashok K. Agrawala was born in India on June 28, 1943. He received the B.Sc. degree from Agra University in 1960, the B.E. degree in electrical technology, and the M.E. degree in applied electronics and servomechanisms from the Indian Institute of Science in 1963 and 1965, respectively. He also received the A.M. and the Ph.D. degrees from Harvard University, Cambridge, MA, in 1970.

From 1968 to 1970 he was a Senior Engineer in the Applied Research Department, Honeywell, Inc., Waltham, MA. In 1970 he joined the Honeywell Information Systems as Principal Engineer in the OCR Department. Since 1971 he has been on the Faculty of the Department of Computer Science, University of Maryland, College Park, where he is currently a Professor. He has been actively involved in research in several aspects of computer systems. He introduced the use of clustering for characterizing the workload of computer systems. Recently, he developed techniques for transient analysis of queues. These techniques have been used to attack several unsolved problems in the design and control of multiprocessor systems, which is an area of his current research interest. In the past, he has published on microprogramming, computer architectures, local area networks, and patterns recognition. He is the author of many papers and two books.

Dr. Agrawala is a member of the Association for Computing Machinery and Sigma Xi.

Ulrich Herzog received all of his degrees in electrical engineering from the University of Stuttgart, Stuttgart, Germany. In 1964 he joined the Institute for Switching Techniques and Data Processing, University of Stuttgart, working in the area of telephone switching systems, data networks, and teletraffic research. In 1973 and 1974 he took a leave of absence to join the Teleprocessing System Optimization Group, IBM T. J. Watson Research Center, Yorktown Heights, NY, mainly being concerned with problems of traffic flow in complex data networks. Since 1976 he has been a Professor at the Institute for Mathematical Machines and Data Processing, Department of Computer Science, Friedrich Alexander University Erlangen-Nürnberg, Erlangen, West Germany. Since October 1980 he has been head of a new chair on computer architecture and performance evaluation. His current research and teaching interests are architecture and performance evaluation of computer systems and networks. As a project leader, he is particularly involved in the design, implementation, and application of the Erlangen General Purpose Array (EGPA), a hierarchically organized multiprocessor computer system.

Dr. Herzog is Vice Chairman of IFIP Working Group 7.3 on “Computer System Modeling.” He also is a committee member of Committee 6, “Technical Computer Science,” of the NTG (German Telecommunication Society) and of the Common Interest Group, “Measurement, Modeling, and Performance Evaluation” of the NTG and GI (German Computer Science Society).