Preface

Computer systems have traditionally been designed with the goal of maximizing the use of the central processing unit (CPU) foremost in the minds of the architects. This approach has resulted from the great expense of designing and building this component of computer systems. Over the years, the trend has been to construct larger and more expensive CPUs, to provide hardware and software constructs to maximize CPU utilization, and to share these resources among many simultaneously executing programs. This was assumed to result in the most cost-effective solution to computational requirements and, incidentally, to provide a means for customers to acquire more powerful computer systems. This strategy has resulted in the design of the large multiprogrammed system complexes of which the IBM Enterprise System and System/390 architectures are recent examples.

While such traditional systems provide cost-effective solutions to a wide range of applications, microprocessor-based computer systems are becoming increasingly important. Integrated-circuit manufacturing technology is providing computer designers the ability to reduce the cost of computer hardware by creating elements for computation, storage, and communication of ever-increasing capabilities and ever-decreasing cost. The concern to maximize the power of a single CPU is no longer as important as it was, and the use of tens, hundreds, or thousands of CPUs in the design of computer systems has become practical. This has reversed the traditional thinking in computer architecture and has led to an increasing amount of research and development in the area of the design of parallel computers and in the necessary system software and the applications to which they can be applied.

At the same time, much work is taking place to enable very powerful computers to cooperate in the more rapid solution of very large problems.

Significant technological problems and the cost of technology prevented the practical realization of the first attempts to build parallel systems (e.g., Illiac at the University of Illinois in 1964). However, by utilizing the newer LSI technology of the 1980s, embodiments of parallel systems have been achieved, and the ability of computer hardware systems to achieve both performance and cost/performance goals without fundamental limitation due to computer hardware has been demonstrated. The emphasis has shifted to increased work on operating systems and environments, compilers and languages, and applications. The more general aspects of parallel computation algorithms and theoretical studies of performance are also of great interest.

Current research and development activities have resulted in numerous product announcements from many new corporations whose goals are to exploit parallel computation. They have also resulted in enhancements to current products in order to take advantage of the new opportunities in parallelism. Laboratories, in the development of future products, are exploiting ever-increasing amounts of parallel computation, to increase both the throughput and computational power of systems. The most fertile area in which new computer architectures and techniques have traditionally been exploited has been scientific computation, and large parallel computers are no exception. The government-sponsored multi-agency program "Grand Challenges: High Performance Computing and Communications" is focused on scalable parallel designs.

These are the concerns of this issue of the *IBM Journal of Research and Development.* The issue reports on a small part of the research activities and results obtained in parallel computation over the past few years in IBM—from laboratories and research centers worldwide, in areas ranging from research in the theory of computation to current products. This should indicate the breadth and depth of upcoming parallel computer systems.

The parallel processing papers in this issue of the *Journal* have been divided into four sections (although several of the papers fit into more than one category).

In the first section, Architectures and hardware, the paper by Shea et al. describes a 256-processor computer (Victor) and summarizes the performance of this parallel system on a large number of real-life problems. The paper by Shimizu et al. is about a multiprocessor comprising off-the-shelf components and the considerations involved in designing its cache mechanism. The paper by Franaszek et al. is a theoretical study of a model of a network for interconnecting processors and memory units in a parallel processing system.

The second section, Operating systems and environments, includes a paper by Bryant et al. that describes the RP3, another highly parallel system, and the operating system concerns for such systems, as well as a paper by Kimelman and Ngo about a valuable graphics system designed for visualizing the dynamic hardware and software behavior of parallel processing systems such as the RP3. (The RP3 and Victor projects were the source of several papers in this issue.) Also in this section is a paper by Ammann et al. describing a system comprising interconnected System/370 microprocessors and the modifications to the 370 hardware and the VM/SP operating system that were necessary for parallel processing, and a paper by Scarborough et al. that discusses the modifications to hardware, operating system, and programming languages that allow cooperative processing among mainframes.

The next group of papers, Applications and analysis, includes a paper by Lorie et al. that investigates how distributed systems with message passing can be used to

speed up the processing of complex database queries, a paper by Reuter et al. that presents an algorithm for vectorizing and parallelizing a frequently used computation on large matrices, a paper by Johnson and Zukowski on algorithms for the exploitation of parallelism in the simulation of massive electronic circuits, and a paper by Flatt that presents a model of performance for parallel systems and provides insights into limitations on speedup.

The fourth group of papers, Languages and compilers, includes a paper by Canetti et al. describing both an extension of the C programming language for parallel processing systems and a prototype compiler, a paper by Hummel and Schonberg dealing with efficient scheduling of fine-grained parallel tasks found in nested parallel constructs, a paper by Ching and Ju describing both techniques for parallelizing ordinary APL programs and a prototype compiler and run-time environment, and a paper by Sarkar that presents an algorithm for optimally partitioning a program into parallel tasks, taking into account the various overheads entailed.

The results reported in this issue form the basis for a new generation of computer systems that has the potential for solving problems with orders of magnitude greater complexity than today's applications and for driving the cost of computation down by orders of magnitude on a broad range of applications, as well as opening the doors to many new applications. In addition to the activity described in this issue, there are several other leading-edge IBM projects in the area of parallel processing architectures: GF11, Many 370, ACE, and the work of the IBM Highly Parallel Supercomputing Systems Laboratory, to name a few.

I thank the authors of all the papers for their fine contributions to the first issue of the *IBM Journal of Research and Development* devoted to parallel computation. I am also in great debt to Dr. Winfried Wilcke, who was my collaborator in the initial conception of this issue and in the early work of soliciting and evaluating papers. His contributions were substantial.

Walter Kleinfelder

Supercomputing Research Center
Bowie, Maryland
(formerly of the IBM Thomas J. Watson Research Center)