

Preface

From its inception, the IBM System/370 family of processors has provided an evolutionary architecture designed to meet the needs of customers with demanding workloads for both commercial and engineering/scientific applications. This issue of the *IBM Systems Journal* focuses on the advances in technology and design that have been implemented in the IBM 3090 system and the System/370 Vector Facility. Additional articles highlight methodologies for improving usability and availability in software systems and describe research in the productive interaction of workstations and main-frames.

When a new system is announced, the immediate focus on improvements in system price-performance often overshadows the underlying advances in technology and architecture. The article by Tucker presents a look at the wide spectrum of new developments in the IBM 3090. The technology shows continuing advances, including the recent announcement (subsequent to the completion of the article) of storage using circuits with over one million bits per chip. Among the features of particular interest to *Journal* readers should be the new error-correction algorithms (that result in the interesting oxymoron "correctable uncorrectable errors"), the implementation of expanded storage supporting synchronous paging without processor interrupts, and the new storage control element that efficiently supports multiple central processors.

Technology and design advances occur at differing rates for the many components that comprise a system. For example, improvements in processor performance have outpaced performance gains in DASD subsystems for the last decade. The paper by Singh, King, and Anderson provides a perspective of how balance has been achieved among the components of the IBM 3090 in providing high levels of total system performance. Several metrics of performance are discussed, along with benchmark results in MVS and VM environments.

When the System/360 was introduced, a major goal was to provide an economical solution to the converging needs of the commercial and scientific users. During the ensuing two decades, while specialized application systems have continued to persist and many applications are finding solutions using workstation computing, the IBM 3090 has provided a new base to economically service the needs of resource-intensive computational applications.

Many engineering and scientific problems need a working combination of substantial computational, very high-speed data delivery and networking capabilities, and large-capacity data storage and management resources. With increasing frequency, these requirements are of an interactive nature and can be best addressed by means of a general-purpose time-sharing environment.

The paper by Gibson, Rain, and Walsh describes the capabilities of the IBM 3090 for engineering and scientific uses. The article highlights the expanded capabilities of the IBM 3090, including the following: greater levels of performance in scalar and parallel processing, new capabilities in the form of System/370 vector processing, very large storage, and extensive programming support routines. Applications benefiting from the new features of the IBM 3090 are discussed.

The IBM Vector Facility operates upon arrays of numeric data stored in main memory, utilizing additional processor instructions and registers. The article by Buchholz describes the new vector architecture that provides enhanced performance while maintaining familiar data, instruction, register, and exception-handling designs.

The IBM 3090 Vector Facility was designed to provide improvements in large-scale engineering/scientific applications, allowing the user to build upon the existing base of software and programming resources available on an established high-

performance system. Complete applications will necessarily vary in their ability to take advantage of the capabilities of a vector processor; they also have processing needs beyond those involving their fundamental computations. Clark and Wilson present design factors that affect the performance of the Vector Facility, as well as factors that allow it to function well in the IBM 3090's storage hierarchy—including effective use of cache and virtual storage.

Ascertaining the usability of software has become a key concern of developers and users of products in virtually every segment of the marketplace. The acceptance of and breadth of interest in a product hinge on its effective usability; indeed, the cost of training may exceed the price of the product itself. Achieving usability becomes particularly difficult for the developer when the software must serve two masters: the dedicated specialist and the occasional user. The former demands maximum flexibility, and the latter requires the simplicity necessary to accomplish tasks with a minimum investment of time. Both are very influential in their own right.

The paper by Gottschalk describes a methodology useful for testing the usability of software products as well as aiding developers in designing for usability. The related paper by Percival and Johnson provides an understanding of the testing approach that has been implemented at an IBM site to study usability.

Unattended operation—particularly during the night and weekend shifts—has become the norm for mainframe computers serving as departmental or local-facility resources. Although there may be no need to attend to machine room peripherals, there remains a requirement to satisfy demand for network and software intervention activities on behalf of terminal users. Advances in the ability of the system to look after these needs are demonstrated in the article by Koved describing Auditor—a VM/370 system monitor that diagnoses faults and initiates service restoration.

Support for the interconnection of workstations and mainframes has developed along many fronts. The article by Kravitz, Lieber, Robbins, and Palermo describes the research project that led to the product PC/VM Bond. PC/VM Bond permits a personal computer (connected to VM/CMS with an IBM 3278

Adapter Card) to transfer messages through the host, emulate an IBM 3278 Display Station, and use CMS storage as a disk server. This project also led to the development of a PC version of the REXX language.

Gary Gershon
Editor