Large systems and Enterprise Systems Architecture

by B. R. Aken, Jr.

A number of diverse factors have influenced the development of IBM's Enterprise Systems Architecture. They range from the compatibility and migration considerations so important for preserving customer investments in existing applications and data, to new functional and capacity requirements of our customers, to the implications of emerging technologies and the projection of these into the systems environments of the future. This paper provides an introduction to a collection of technical papers in this issue describing the ESA/370™ facilities. Its purpose is to convey a broad perspective on important factors that will influence the large-systems environment of the future, and to relate those factors to the key elements of the ESA/370 architectural enhancements. It does not attempt to address all of the considerations leading to the development of ESA/370, nor discuss the new features in any depth. Detailed discussion of the specific features, facilities, and design considerations of ESA/370 will be found in other papers in this issue.

The year 1988 was significant in the evolution of IBM's large systems. It was marked by the introduction of the Enterprise Systems Architecture (ESA) and the new Enterprise Systems Architecture/370™ (ESA/370™) family of IBM systems and supporting products. ESA represents a significant milestone in the evolution and focus of IBM's large systems. It establishes important architectural and system software foundations that are essential in meeting the emerging requirements of today's large-system customers and are crucial to the role evolving for the large systems of the future.

This issue of the *IBM Systems Journal* has as its theme large systems and Enterprise Systems Architecture. Its goal is to provide the reader with a perspective on the evolving role of large systems and to provide an in-depth understanding of the ESA facilities as they relate to that evolution.

A number of diverse factors have influenced the development of ESA/370. The desire to preserve investments in existing programs and data is certainly a primary one. Such compatibility goals clearly tend to limit the form and degree of change desirable or practical. At the same time, the requirement to rapidly bring technological progress to the marketplace, and the emerging customer requirements for new function and large-scale data processing capacity that result, strongly suggest the need to make significant architectural strides forward. These often opposing forces require a balanced and very thoughtful approach to the evolution of system architecture. ESA/370 represents the culmination of years of effort to achieve both a high degree of compatibility and a major step forward in key aspects of system architecture.This paper provides an introduction to a collection of technical papers describing the ESA/370 facilities. Its purpose is to convey a broad perspective on important factors influencing the large-systems environment, and to relate those factors to the key elements of the ESA/370 architectural enhancements. It does not attempt to address all of the considerations leading to the development of ESA/370, nor does it discuss the new features in any depth. Detailed discussion of the specific features, facilities, and design considerations of ESA/370 will be found in the subsequent papers of this issue.

Large systems

To better understand the evolution occurring in large systems, it is helpful to have some perspective on the history and evolution of these systems to date. Large systems and the computer industry in general have undergone a process of continual evolution. The very terms *large system* and *large-systems marketplace* have become less definitive through the years. There is not some well-defined point beyond which a system is "large" and below which it is not. Nor is there a clearly defined marketplace in which only "large systems" can participate.

The view of what constitutes a large system changes with time and in any given era is a function of the technology available at that time. A large system tends to be thought of as one built with leading-edge technologies of the era and positioned near the high end of the range of single-system computational and data storage capability generally available in the marketplace. Thus, the capabilities of these systems increase with each new generation of technology. The term *mainframe* is also generally applied to these systems.

Of course, those same advances in technology also lead to increased capabilities for smaller systems built with more "off-the-shelf" technologies. Today there is a profusion of system classification terminology. Words like microcomputers, personal computers, intelligent workstations, mini-computers, superminis, minisupers, midrange, mainframes, and supercomputers reflect the growing terminology and specialized focus areas for computer systems. While it is clear that many of these are not thought of as large systems, it is just as clear that classes of large computing problems, which might traditionally have required a large system to solve, are being solved today with systems that are not considered to be large. Smaller systems are being specially designed to deal with specific classes of large problems very effectively. They are also being interconnected to provide larger distributed computing services which provide many services comparable to those available from a single larger system. The classification of systems is quite imprecise, and the boundaries of the relevant marketplaces are even less precise. The continuing advances in technology will only make this blurring more pronounced.

Nevertheless, one can usefully characterize the classical large-system environment and discuss the changes being wrought by technological progress. The classical large system, in addition to being a large-capacity, technologically leading-edge system, is also generally characterized by a "central-system" structure as opposed to a distributed one.

The general model of such a system is that of the large computer or data center. These centers contain one or more large computers, very large quantities of data stored on direct-access and tape storage devices, large collections of applications, and a number of high-speed print devices to handle the large quantities of printed materials typically generated. There are hundreds and even thousands of users of a large facility, and these users are typically connected via some form of nonintelligent display station such as the IBM 3279. A control program such as Multiple Virtual Storage (MVS) or the Virtual Machine facility (VM), in conjunction with other system software products, is responsible for controlling the use of system resources. These products provide higher-level services, such as terminal handling, application scheduling, and database management, and also have responsibilities for managing the efficient use of system resources and protecting the information assets of the facility from unauthorized access.

These data centers are often linked with other data centers in large network configurations, but fundamentally each data center is an independent, autonomous facility. The central system is always managed by a computing center staff whose job it is to operate, maintain, and administer its facilities. Negotiations with this staff are required whenever usage requirements change.

The data and services available to a user of a centrally structured system tend to be bounded by the domain of the system or data center to which the user is connected. Thus, the recurring theme of the classical large system is "central"—central processing, central data, central applications, central management.

The central system contrasts with a distributed systems structure where multiple systems are interconnected with processing, data, application, and control facilities distributed in some fashion among the systems. Many of the systems might be single-user workstation-type systems, and others might perform specialized roles such as data servers, network gateways, or shared application systems. The facilities available to a user in a distributed system structure are not bounded by the specific system to which the user is connected. Instead, they tend to be bounded by the facilities of the network and the access rights of the user.

The investment associated with large systems has always resulted in a strong interest to maximize their effective utilization.

With the very nature of large systems being large, leading-edge systems, they have always represented a significant investment. The cost of early large systems ranged from hundreds of thousands to millions of dollars and today's large systems still range from hundreds of thousands to several million dollars, though their processing and storage capabilities have increased manyfold. The large-systems marketplace has, therefore, been primarily that of big corporations, government, education, research, and other large institutions. These organizations have had the large-scale computational requirements, the facilities, and the assets necessary to purchase and profitably utilize such systems. The primary computational tasks in this marketplace have been those associated with general-purpose business information processing. These tasks encompass a wide variety of applications such as payroll, accounts receivable, accounts payable, inventory control, customer master files, information databases, publications, credit/debit processing, electronic funds transfer, airline reservations, complex network scheduling (communications, railroads, seaports), manufacturing control, engineering design, and a plethora of other industry applications. The systems are also heavily used for interactive personal computing, program development, office applications, and a variety of scientific applications such as geological data mapping, modeling, simulation, and computer-aided design. Many of these applications can run concurrently on a single system. The systems have general applicability to a very wide variety of uses, and much of the world's corporate and institutional business is conducted on them today.

The large, very visible investment associated with large systems has always resulted in a strong interest on the part of the customer to maximize their effective utilization. Customer requirements have been dominated by the desire for both cost reduction and the ability to scale up, or "grow," their business applications. The inability to grow a business application is an inhibitor to growing the business supported by that application. The lack of available capacity can also impede efforts to develop new applications. If the largest system available is completely utilized by a single growing application, major costs and schedule delays may be experienced while the application is restructured to improve its efficiency, or the data and applications are somehow partitioned to allow multiple systems to be utilized. Business growth can be gated by such activities, and large, complex applications can take years to redo. Such activities are generally undesirable from a business perspective as they require investment to redo something that already works satisfactorily, except for applying more computing power to it. It is much more desirable to invest in activities that introduce new products or services which result in new and additional business revenues.

Restructuring business applications for purely growth reasons can also result in ongoing additional business costs when such restructuring results in the disintegration of business data or applications. Applications that could readily share data on the same system very efficiently might be forced into less optimal sharing techniques across multiple systems. Or, data which was previously only required on a single system might need to be split apart, or maintained in replicated copies across multiple systems with attendant procedures to keep the copies in synchronization with one another. Whole classes of transaction processing applications requiring high-volume processing can become impractical when the data to be accessed is not local to the system supporting the application.

The focus of much of IBM's large-systems evolution has therefore been oriented toward rapidly bringing new hardware and software technologies to market to continue to lower the cost of information processing and maximize the ability to scale up existing business applications. Operating systems such as MVS and VM, with their attendant system products such as Information Management System (IMS), Customer Information Control System (CICS), Conversational Monitor System (CMS), and Database 2 (DB2), have evolved through the years with a strong focus on internal efficiency, the ability to take advantage of available system resources, and the ability to be effective over a broad range of problem scale.

Common themes of product announcements, in addition to the introduction of new technology, have been the raising of architecture or design limits, and the improvement of algorithms in areas of efficiency, parallelism, responsiveness, throughput, control, or behavior in the face of increasing scale. It would not be useful to try to summarize here the long litany of product evolution that has occurred over the last two decades relating to such characteristics. But it should be useful, and relevant, to characterize a few of the key design parameters that require continuous reevaluation as technology and problem scale change.

One of the major design factors in large systems is the very wide imbalance in the time scale that exists across the different system elements. One can understand this situation by simplistically thinking of the computer system as being comprised of three basic types of devices: instruction-processing devices (the computation units and main memory); data storage devices (disk, tape); and end-use devices (devices to get information into and out of the computer, such as keyboards, displays, printers, and card readers). The technologies used in each of these devices are quite different. The computational and main memory elements are built of electronics with no mechanical motion and can therefore be made to perform extremely rapidly. Most bulk data storage and end-use devices involve mechanical motion and communications protocols that operate at significantly slower rates than the instruction-processing units. On the time scale of modern computers, a person sitting at a terminal takes an unbearably long time to complete the simple task of striking a key.

As an example, a person modestly skilled at typing might type at a rate of 30 characters per second. In that same second, a single processing element of the IBM 3090 Model 600S processing system might process 15 to 20 million instructions. That is a half to two-thirds of a million instructions processed in the time a person takes to type a single character.

Similarly, accessing stored data is also very slow compared to instruction-processing speeds. Obtaining a record from an unbuffered disk storage device might take on the order of 25 milliseconds (thousandths of a second). In that same amount of time, a 20-million-instruction-per-second processor could execute a half-million instructions. Many transactions require fewer than a half-million instructions to be processed in their entirety.

If the processing system had to wait for input/output (I/O) operations such as these, a tremendous percentage of the computing potential of a large system would be wasted.

Even personal computers are fast in relation to I/O operations. Utilization of their full processing potential is typically quite low, even when they are being actively used. However, as the cost of an item declines, the concern for the degree of utilization of it also declines. We do not generally concern ourselves with the degree of utilization of pencils and calculators when making a purchase decision. Other factors such as convenience, functionality, usability, and even styling become more important as cost declines.

The high cost and the sheer scale of large systems make it essential that they be very effectively utilized, and techniques for improving utilizations receive unending attention. One of the primary techniques for improving utilization is multiprogramming.

The concept of multiprogramming is somewhat analogous to juggling. While waiting for a single ball to rise and fall, the juggler can grab several more balls in succession, throwing each one up in the air for about the same amount of wait time, thereby keeping them all in motion without dropping any. Similarly, a control program can load multiple programs into the computer memory, and when the currently executing program makes a request that requires waiting (such as obtaining some data from disk storage or getting a screen of input from a display), the control program can save the state of that program and switch to working on another that is not waiting for anything to complete. The use of the processor can thereby be overlapped with the waiting for slower I/O operations. Multiprogramming significantly improves the utilization of the resources of the large system, allowing much larger workloads to be processed per unit of time.

A control program such as MVS with its related subsystems will have hundreds or thousands of programs in varying stages of execution at any instant in time, and depending on the nature of the tasks, such a system might process millions of user transactions per day. Small changes in multiprogramming efficiency and control can have significant influence on overall system performance and throughput.

Multiprogramming significantly improves the utilization of the resources of the large system.

A second major approach to improving the efficiency of large systems is to utilize algorithms or techniques which shorten the wait time within programs. A significant amount of wait time is associated with the I/O operations needed to gain access to stored data. Data buffering and data caching techniques have therefore been developed either to anticipate the need for data in advance and bring the data into high-speed memory before the program actually needs it, or to keep data that has been recently referenced in high-speed memory longer than is necessary in anticipation of its being referenced again soon. Processing a sequential file of customer records is an example where the next data reference of a program can be anticipated on the basis of its last one. Database index files and program directories are examples of data that are likely to have high reference rates.

There are many design trade-offs to be considered when performing optimizations on systems as complex as these. For data caching techniques one needs to consider the amount of memory that can physically be attached, the higher costs of high-speed memory, the value of alternative uses of the memory, and the effectiveness of the available buffering algorithms for any given workload. But minimizing program wait time is generally well worth some added effort, as it has several positive effects. It reduces the number of programs that must be kept in memory simultaneously in order to fully utilize the system resources (memory overhead); it reduces the amount of system resource required for switching between programs (processing overhead); it improves the responsiveness of the system to its users (shorter wait times); and it minimizes the time that programs hold shared resources, thereby minimizing the probability of contention for those resources and further reducing wait times.

Data sharing is a cornerstone of many of today's important large-systems applications. This is especially true for high-performance transaction processing applications developed for the IMS and CICS transaction processing subsystems, but it is equally true for the rapidly growing relational database applications based on the DB2 and Structured Query Language/Data System (SQL/DS) products. The performance of these application environments can be dramatically affected by small changes in data access performance, system overheads, or resource hold times. Thus, the clever use of high-speed electronic storage for improving the performance of these systems has been an ongoing focus of system design.

Large corporations and institutions have developed extremely large and complex information systems based on today's technologies. Large installations will generally have multiple large computers. They have hundreds of large-capacity data storage devices (literally occupying acres of floor space) with thousands of data files and databases encompassing trillions of bytes of data. The storage devices will generally be connected to multiples of the computers through complex switching devices to allow sharing of the data and to allow continued access to the data in the case of system failures. The ability to grow these databases and efficiently process the data within them is a significant competitive factor. As a result, it is also a significant business opportunity for the large-systems vendor. Changes in system architecture or design that can significantly improve the performance and scale-up characteristics of the systems are continuously sought.

The continuing scale-up of any system, be it a computer system, a biological system, or simply an organizational system, requires ongoing attention to the appropriateness of the underlying architectural assumptions upon which it is based. Periodic, and sometimes fundamental, change is mandatory for systems to remain efficient in the face of continued growth and changing technology. The continual evolution of IBM's large-system architecture and system products from System/360 through 370-XA over a period exceeding two decades reflects an ongoing attention to the need for system architecture and system software to evolve. The continuing evolution of these systems in the face of today's rapidly evolving technology was again a driving factor leading to the development of ESA/370.

The evolving large-systems environment

In addition to the traditional requirements for evolutionary bigger, better, and lower-cost systems, other factors have also had significant influence on the evolution of large systems and the development of ESA/370. As might be expected, technological change is again the primary catalyst.

The last two decades have seen tremendous advances in technology and a flood of exciting new products resulting from it. This steady flow of technological progress has resulted in major enhancements to the capabilities and capacity of large computer systems, as well as a stampede of lower-cost, smaller systems of all types. The cost of computing has come down to the point where quite sophisticated personal computer systems are virtual commodity items today. Such technological advances have profound implications on the structure of computing systems.

Many of the design choices still visible in today's large systems have historical roots in a period of time when technology costs were much different than today. In the 1960s when System/360 was being conceived, computational, storage, and print facilities were very expensive resources, as well as being environmentally constrained in the sense of size, power, and cooling requirements. Centralizing the use of this technology was a very optimal thing to do, as it minimized the amount of expensive technology required and made the systems affordable to a broader marketplace. Such considerations strongly influenced the central-system structure that still tends to characterize the large mainframe systems of today. For example, the terminal device with which a user accesses the computer was designed to have virtually no processing intelligence. It would merely perform a series of simple commands as directed from the central computer. The central computer had all the intelligence as to what to display, where to display it, and what to do in the case of errors. Therefore, the terminal could be built as a relatively low-cost device, and more users would be able to have access to the computer. Similarly, storage devices were built to have a minimum of local processing intelligence so that their costs could be minimized and data storage space would be more affordable. Applications and data were managed and stored at the central system, and operating systems emerged to tackle the complex tasks of controlling the processor and devices and managing user access to the applications and data.

Computers are increasingly being bought because an application is available for them.

The cost of providing "intelligence" in devices today has been reduced to the point of near irrelevance. Whereas a computer was a very expensive resource in the 1960s, you can walk into many a shopping mall today and purchase a complete personal system for a few hundred to a few thousand dollars. You can carry it away, plug it in anywhere, and begin productively using it almost immediately. These systems even come in laptop versions which operate on battery power. With the addition of a relatively inexpensive communications adapter and some software, the personal system can communicate and cooperate with other computers virtually anywhere in the world.

Computers are becoming pervasive in the marketplace, and there has been an explosion of easy-to-use personal, business, and scientific applications developed for these systems. Computers are increasingly being bought simply because an application is available for them. It is often more cost-effective to buy an application and a separate computer to run it than to negotiate with a central system support staff for the capability to install and use a comparable application on a mainframe. Mainframes have a tendency to already be heavily loaded with critical business applications.

But even more important, these personal computers can also provide highly responsive, user-friendly forms of interaction. The personal computer is continuously attentive to the keyboard, and information can be placed on its display screen at internal memory speeds rather than communications line speeds. Action bars, mice, roller balls, pop-up/pull-down menus, icons, animation, real-time graphics, touch screens, and voice synthesizing are some of a list of user-friendly, highly interactive interfacing techniques that continues to grow. Many of these techniques are simply impractical when controlled from a remote large system. The communications delays erode the responsiveness characteristics for the end user, and the need to be highly attentive to a large number of users would result in unacceptable multiprogramming system overheads in the large system.

One of the implications of this technological progress is quite clear. The personal computer, or intelligent work station (IWS), will increasingly become the dominant interface to the end user. These systems are also capable of performing sophisticated user-application functions today, and their local processing and data storage potential will continue to increase as improvements in technology show no signs of slowing down. Local user applications can be expected to track this technological progress, becoming increasingly sophisticated as the power of the IWS increases. Many end-use applications and large quantities of stored data will either exist at these systems or be readily accessible via service machines in local-area network type configurations. The central-system model will certainly not be the model that these end users will have of the computing services available to them.

A second significant factor influencing the evolving role of large systems is the growth in the quantity of data being captured on line and the inherent business value in performing more complex operations against it and extracting higher order information from it.

There is inherent value in lower-cost data processing. It can be used to extract valuable new levels of information, provide more timely information, reduce the cost or duration of activities, and so on. Corporations use this technological progress to lower their business costs, improve their service to customers, and introduce new revenue-producing services that are made possible by the improvements in technology. To stand still and not move forward with technology would be to become stagnant and uncompetitive.

The classical large-system applications continue to exist, but the sophistication and the scale of these applications tend to grow as technology makes new forms of processing viable, and as the parent enterprise expands its customer and services base. In addition, entire new classes of applications emerge each time technology significantly lowers costs and increases the capabilities of the systems.

Behind these applications is an ever-growing collection of data which continues to become increasingly more fundamental to the very operation of the enterprises. Customer files, employee files, payrolls, inventories, revenues, expenses, product specifications, design databases, credit files, process control data, research studies, document libraries, phone directories, patent portfolios, competitive analysis data, market trend data, demographics, mailing lists—the databases continue to increase in numbers, size, complexity, and diversity as it becomes possible to capture the information and profitably process it. Processing power can look for new relationships and trends in growing bases of data and assist in making more timely decisions. Emerging technologies such as digital imaging and digital voice recording further explode the quantities and forms of data that can be readily captured. Information is the goal of information processing, and information is extracted from data.

The trend in society continues toward a more service-oriented economy.

Entire new classes of applications are emerging based on increased volumes of on-line data and higher levels of data and application integration. Computer-aided design (CAD) and computer-integrated manufacturing (CIM) applications allow an engineer to interactively develop designs on high-resolution graphics workstations and move those designs directly into manufacturing in a fraction of the time required with traditional techniques. Desktop publishing packages and laser printers allow virtually anyone to produce high-quality, multifont publications with integrated text, image, and graphics. Integrated office products bring facilities for electronic mail, time management, document creation, document filing, meeting scheduling, and decision support systems together into a single cohesive set of applications supporting the day-to-day tasks of the business professional and connect everybody into the extensive information processing and network facilities of the larger enterprise. Digital imaging of such items as personal checks, credit card slips, insurance forms, or library documents has the potential to provide tremendous cost savings by reducing paper handling, eliminating lost records, and allowing electronic processing, mailing, and filing. Artificial intelligence offers an entire new field of applications.

The trend in society in general continues toward a more service-oriented economy, and this reflects itself in the business community. Home banking, automatic teller machine (ATM) services, videotext, computer shopping, financial market services, polling services, on-line travel services, and on-line library services are all indicators of the type of service-oriented economy emerging. Tremendous business opportunities are associated with it. The trends continue to bring more data on line, keep that data more current, and require broader, more responsive access to the data in order to derive higher-quality, more valuable information from the data.

New technologies for capturing information not only increase the sources of data and the rate at which data can be captured, they also increase the density of the captured data. Where a record of a personal check might require only a few hundred characters (or bytes) of text information to be represented, a digital image of a check requires on the order of 50 thousand bytes of information. Similarly, the data content of digital voice recordings is considerably more dense than the equivalent character representation of the recorded message. Such increases in data density result in both increased storage capacity requirements and increased processing and data manipulation requirements.

As the size of data elements increases, the importance of minimizing I/O processing delays also increases. The data representation of images can often be significantly reduced in size by applying data compression algorithms. Thus, it also becomes important to consider utilizing some of the processing power of the computer to compress and decompress images to reduce the quantity of storage required and reduce the amount of information that must be transferred between system elements.

These trends in data capture and information processing growth show a continued business opportunity for the highest-performance, largest-capacity information storage and processing systems that technology will allow.

These combined factors point to one of the major emerging opportunities for the large systems of the future: that of the high-performance, large-application and data-service facility in an integrated information processing network. Although many of the services provided by today's large systems, such as high-volume transaction processing, background job scheduling, or central management and control of assets, will continue to evolve on these systems, many of the personal-use and less-demanding applications (from the point of view of not requiring a large system) will be performed outside of the large system. But significant new business opportunities and requirements will exist for connectivity from the growing population of intelligent workstations and other network-connected systems to the large-application and complex data services that continue to evolve from today's large systems. The key to enabling this opportunity will be the connectivity and cooperative processing capabilities of the large systems, as well as their ability to provide the very high-performance, complex data services cost effectively. The ESA/370 facilities are directly focused on the latter of these two areas.

Enterprise Systems

The Enterprise Systems announcements of 1988 are focused on a set of evolutionary steps to provide very large, cost-effective, complex data- and application-server systems. The hardware system architecture and capacity enhancements of the ESA/370 systems are the foundations for the software system products that were announced and are yet to follow.

The major enhancements are in the areas of memory addressability, program linkage, storage hierarchies, system-managed storage, and system logical partitioning. Also very important is the continued increase in the quantities of electronic storage that can be attached to the systems.

The importance of these facilities can be clearly related to the earlier discussions of areas of continuing focus in large-systems design. One of the prime factors limiting the ability to take maximum advantage of large-systems processor performance enhancements is the relatively slow technologies available for access to stored data. Significant processing potential is spent switching between tasks while waiting for data to arrive in memory. Techniques that cache or buffer data in high-speed storage have been effective in reducing system overhead for selected classes of data, but the quantities of electronic storage available have been limited, and the bulk of data requests are still satisfied today from relatively slow direct-access storage devices.

Significant processing potential is spent switching between tasks while waiting for data to arrive in memory.

With technology advances continuing to improve the cost and density of electronic storage, the possibility of using very large electronic memories either directly as data storage devices, or configured as part of a system-wide storage hierarchy, emerges as a cost-effective technique to significantly improve data access times. The ESA/370 Model 600S processing system can attach 512 megabytes (million bytes) of byte-addressable main memory and 2 gigabytes (billion bytes) of electronic block-addressable memory, thereby providing significant opportunities in this area. The use of electronic memory for data access has the potential to significantly increase the transaction processing rates sustainable by these systems as well as provide higher levels of responsiveness to large, relational data operations. The paper by Cohen, King, and Brady[1] in this issue discusses the subject of storage hierarchies in depth.

The ability of programs to efficiently manipulate large quantities of data in memory will also be very important for these systems, and ESA/370 includes significant enhancements in this arena. The prior 370-XA architecture increased the addressability of programs from 16 megabytes to 2 gigabytes of linear address space. In this architecture, the operands of the instructions for the program all referred to addresses within the address space containing the program. Although special operations and modes were provided to allow data to be moved between address spaces or to allow programs to call programs in other address spaces, the program was basically limited in its domain of addressability to the single 2-gigabyte address space in which it existed.

The ESA/370 architecture provides several important enhancements in the area of addressability. Most importantly, ESA significantly extends program addressability by allowing the entire processor instruction set to operate against multiple address spaces. Each operand of an instruction can point to an object in a different address space. The potential program addressability is thereby increased from that of a single 2-gigabyte address space to 2 gigabytes times the maximum number of allowable address spaces. With the current architecture, this is an increase in excess of three orders of magnitude, allowing for 16 terabytes of addressability.

Secondly, the ESA addressing architecture extensions make possible the concept of a "data only" address space, referred to as a *data space.* The data space becomes a very useful construct for providing higher levels of data object encapsulation and for mapping persistent stored data objects (such as files, directories, indexes, or databases) directly into program-addressable virtual memory. By simply manipulating the addressability tables of the system, entire data objects (or portions thereof) can become directly addressable in the virtual memory of a program.

Thirdly, the ESA extensions include new improvements for interprogram linkages. With ESA, new hardware assistance is provided to automatically stack and unstack program status across program calls and assist in the automatic establishment of recovery environments to deal with unanticipated error conditions. These assists provide further performance and RAS (reliability, availability, and serviceability) improvements for programs using the new linkages and can help reduce multiprogramming overheads.

New system services are provided within MVS to support the new architectural facilities and to enable the mapping of data into the memory hierarchy.

The ESA/370 architecture is described in detail by Plambeck.[2] Additional perspectives on the technical motivations and rationale for the architecture will be found in the paper by Scalzi, Ganek, and Schmalz.[3] The MVS operating system support of the new architecture facilities is described by Clark[4] and by Rubsam.[5]

In addition to the requirements for rapid access to data, large installations are faced with significant

Managing large-storage configurations involves a number of complex tasks.

challenges in managing and controlling their information assets. Large installations today have hundreds of high-capacity direct-access storage devices, and the pace of new data capture seems destined to continue to outpace the technology improvements in storage densities. Managing large-storage configurations involves a number of complex tasks such as performance tuning, database reorganization, migration of unreferenced data to lower-cost storage, maintenance of backup copies of data, maintaining disaster recovery capability, installing new storage devices and removing old ones, protecting data from unauthorized access, and generally administering data access rules. The complexity and time required to perform these tasks tend to scale with the complexity of the configurations and the amount of data being stored. At the same time, a continuing trend toward 24-hour-per-day, seven-day-per-week operations leaves less and less time available to perform the tasks. There are long-standing large-system customer requirements for more automated storage management facilities which never require making the data unavailable to users. The Data Facilities Storage Management Subsystem (DFSMS™) products announced with ESA are a major step in realizing this concept of system-managed storage. A storage-management architecture is established to allow the system to manage the extensive data storage facilities of the installation on the basis of individualized administrative rules appropriate to it. This architecture is integrated with the large data addressability and large electronic memory hierarchy of the ESA/370 systems to provide synergism across the entire collection of hardware and software elements of the data storage facility. The paper by Gelb[6] describes these system-managed storage facilities.

The last major element of the ESA enhancements is the Processor Resource/Systems Manager™ (PR/SM™). PR/SM enhances the system architecture by providing the capability to logically partition a large system into multiple independent system environments. Thus, a single system can act as multiple smaller ones.

There are many reasons why such a facility is useful to customers today, but as the size and complexity of these system configurations continue to grow, it will be increasingly valuable to be able to temporarily configure test, migration, or development systems out of the production system resources rather than be forced to take production systems down, or purchase completely separate systems for those purposes. The paper by Borden, Hennessy, and Rymarczyk[7] describes the facilities of PR/SM.

Summary

In summary, the role that the large, leading-edge systems play in the industry continues to evolve as technological progress continues to occur. Low-cost computing and end-user interfacing technologies are driving significant end-user application and data resources out into the network, whereas new data capture technologies and value-added information processing opportunities are stressing the performance and capacity offerings of the largest data processing systems available. The affect of these evolutionary forces on large systems will be toward shaping large systems into very high-capacity, high-performance application and data servers in highly interconnected peer network configurations. The architectural extensions of ESA/370 with their attendant system product support are foundational in making new strides towards larger-capacity, higher-performance, more cost-effective data server systems to address these emerging requirements.

Cited references

1. E. I. Cohen, G. M. King, and J. T. Brady, "Storage hierarchies," *IBM Systems Journal* **28**, No. 1, 62–76 (1989, this issue).

2. K. E. Plambeck, "Concepts of Enterprise Systems Architecture/370," *IBM Systems Journal* **28,** No. 1, 39-61 (1989, this issue).
3. C. A. Scalzi, A. G. Ganek, and R. J. Schmalz, "Enterprise Systems Architecture/370: An architecture for multiple virtual space access and authorization," *IBM Systems Journal* **28,** No. 1, 15-38 (1989, this issue).
4. C. E. Clark, "The facilities and evolution of MVS/ESA," *IBM Systems Journal* **28,** No. 1, 124-150 (1989, this issue).
5. K. G. Rubsam, "MVS data services," *IBM Systems Journal* **28,** No. 1, 151-164 (1989, this issue).
6. J. P. Gelb, "System-managed storage," *IBM Systems Journal* **28,** No. 1, 77-103 (1989, this issue).
7. T. L. Borden, J. P. Hennessy, and J. W. Rymarczyk, "Multiple operating systems on one processor complex," *IBM Systems Journal* **28,** No. 1, 104-123 (1989, this issue).

Bernard R. Aken, Jr. *IBM Data Systems Division, P.O. Box 390, Poughkeepsie, New York 12602.* Mr. Aken is an IBM Fellow at the Myers Corners Laboratory in Poughkeepsie, New York. He joined IBM in 1966 and began his career in large systems in early 1969. He has held a variety of key technical and management responsibilities throughout the evolution of the MVT and MVS operating systems and continues to have technical leadership responsibilities for the future evolution of MVS.

Reprint Order No. G321-5345.