IBM 3090 performance: A balanced system approach

by Y. Singh
G. M. King
J. W. Anderson

The IBM 3090 system represents the highest level of system performance offered by IBM to date. To realize the full performance potential of this system, it is essential to maintain a balance among its various components. The major components of the system are the processor(s), storage, I/O, and the software that manages the system resources. Their performance attributes are discussed and their effect on system performance illustrated by laboratory benchmark measurements for the MVS and VM operating systems.

The demand for processing performance has grown greatly during the last decade. In the middle 1970s, processing performance requirements were following an annual compound growth rate (CGR) of about 20 percent; during the late 1970s and early 1980s the CGR was about 40 percent; and during the middle 1980s the CGR is expected to be even higher. This growth reflects both the addition of new areas of computer applications and the proliferation of the use of existing applications.

The accelerating growth in performance requirements corresponds to a shift from predominantly batch applications to Data Base/Data Communications and, more recently, to end-user applications. This evolution is a result of the compounding effects of new applications, made economically feasible by continuing improvement in the price/performance of data processing equipment and by expanding terminal networks that provide a larger number of people with access to the applications.

The growth in customer demand for computing power presents both an opportunity and a challenge to the industry. Systems must be designed to allow for a rapid growth in processing power and the other resources required to put it to productive use. Processor performance must support the growth in customer requirements with minimal disruption to operations. The operating system must support and manage growing levels of system performance, using fundamental design algorithms that span a wide range of performance. A system's structural and algorithmic limitations have to be identified early so that solutions can be designed, developed, and delivered to the customers before the effects of these limitations interfere with the customers' ability to satisfy their growth requirements.

This paper first discusses some of the major considerations that influenced the various aspects of system performance of the IBM 3090 during the product design phase. The concept of a balanced system is discussed next, including its various components and their performance attributes, and how this balance has been achieved. Finally, some performance results from laboratory benchmarks for the MVS and VM operating systems are summarized.

A large system must satisfy the customer requirements consistent with the evolution of large systems.

Design considerations

The design point of a large system, such as the IBM 3090 product line, must satisfy the customer requirements in a manner consistent with the evolution of large systems. The success of the product line is determined by how well the diverse and often conflicting requirements are met across a broad spectrum of customer applications. The requirements discussed in this section are for illustrative purposes and are not intended to be a comprehensive list. The final 3090 product characteristics resulted from appropriate trade-offs made at every stage of product design to yield a product capable of satisfying the customer requirements.

Performance. The performance of a system must address customer requirements for information processing. High-end customers have demonstrated a rapid growth in system throughput requirements, and this growth rate is expected to continue, possibly even increase, in the foreseeable future. Satisfying performance requirements calls for the exploitation of advances in semiconductor and packaging technology as well as processor and system organization. In addition to improving performance at the processor level, it is important to ensure that appropriate enhancements are made to the architecture, software, and I/O subsystem to realize the full performance potential of the processor at the system level.

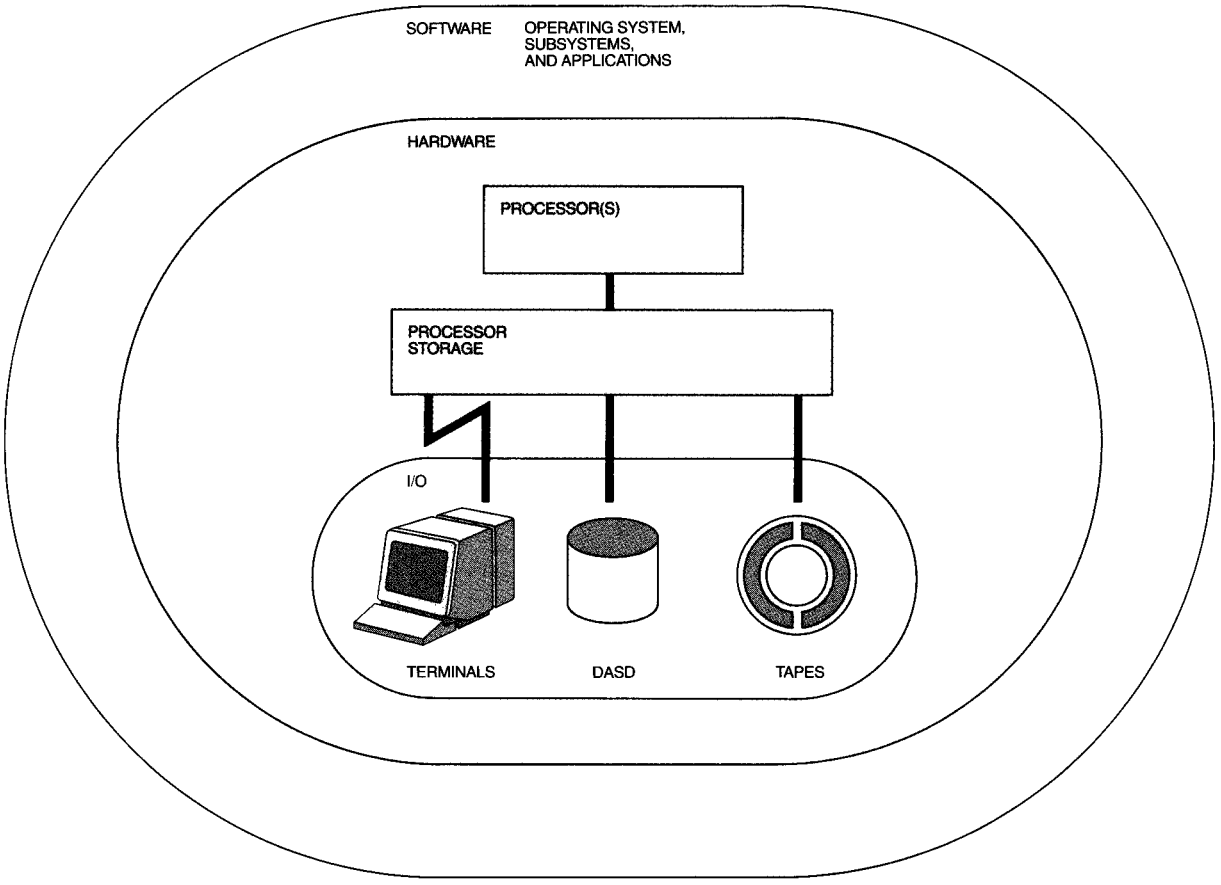
Reliability, availability, and serviceability. A number of customers depend on large systems for efficient information processing to support their operations. Because large systems can process this information rapidly and continuously, these systems have become an integral part of their business operations. During the past few years, customers have shifted from prime-shift information processing requirements toward twenty-four-hour-per-day operation. This has resulted in a need for continuing improvements in the reliability and availability of the system.

Function and architecture. Over the years, IBM large systems have incorporated new and enhanced functional and architectural capabilities in response to customer requirements. These capabilities also support and exploit the ever-increasing performance and configuration choices offered by advances in technology and machine organization. These capabilities have to provide full backward compatibility to allow continued running of existing customer applications while overcoming constraints or providing new capability for new applications.

Upgradability. The processor represents a significant capital outlay for the customer; therefore, it is essential that the customer be able to deploy this asset for an economically reasonable period of time. As system performance requirements grow over time, customers should be able to add to information processing capability incrementally, with increments of reasonable size, in order to satisfy capacity growth requirements without making purchased assets obsolete. IBM offers a solution to this requirement by providing upgradability within the product family (e.g., the IBM 308X system). This upgradability within the product family is fundamental to satisfying growing customer performance requirements while preserving the value of the asset. The IBM 3090 continues to provide this capability.

Exploitation of technology. The IBM 3090 represents a new standard for IBM large-system performance. One of the major ingredients for accomplishing this is the use of sophisticated technology, with speed and density enhancements to realize

Figure 1 Components of a system



the fast cycle time of the machine. It takes the technological advances made with the IBM 308X processor family and extends them very substantially to satisfy the cost, performance, and reliability requirements for the IBM 3090.[1] Technology trade-offs must be made early in the product design phase to yield a machine with the right mixture and balance of different technologies. In addition, product design must retain as much flexibility as possible to take advantage of technological advances that may take place during the life of the product.

Balanced system

Systems must be designed with an increasing capacity to process work. A system, as shown in Figure 1, consists of hardware elements (processor, I/O, and storage) as well as software elements (operating system, subsystems, and application programs). The system design must strike a balance among these key elements, with the software managing and exploiting the resources and capability offered by the hardware elements.[2]

The system processes work requests received from the users either as transactions from terminal users or as large jobs submitted for batch processing. A user perceives system performance as the time spent waiting for the response to a request, whether it is the system response time for interactive transactions or the elapsed time for batch jobs. From the viewpoint of the installation manager, however, system performance may be measured in terms of throughput (such as transactions per second or jobs per hour) within specified response time criteria.

The throughput of the system can become limited when one of the elements of the system is no longer capable of supporting a higher volume of work. Depending upon the nature of the constraining resource, the performance bottleneck may manifest itself through a degradation in expected performance. When feasible, the constraining resource is augmented to remove the bottleneck.

When any element of the system is replaced by one with higher performance capability, the other elements may need to be enhanced.

Just as it is important to maintain an existing system in balance to realize its full performance potential, when any element of the system is replaced by one with higher performance capability, the other elements of the system may need to be enhanced to maintain the right balance among the various resources.

We now discuss the major elements of the system (processor, I/O, storage, and software), their performance attributes, and key concepts/techniques for ensuring a balanced system.

Processor. The processor subsystem consists of one or more instruction processing elements, called *central processors*, coupled together by the system controller and sharing common storage. The processor subsystem runs under a single control program (operating system). The performance of the processors is a function of the individual central processors and the efficiency of their interconnection.

From early in its development, the 3090 was designed to include one Vector Facility per processor. The I/O, storage, and operating system designs were chosen to balance not only the scalar processors but also the Vector Facilities.

The performance of a single central processor is a function of the cycle time and the average cycles per instruction for any given workload. The cycle time is determined by the speed and density of the circuit technology as well as by the amount of function performed per cycle. The IBM 3090 processors utilize sophisticated semiconductor and packaging technology to yield a processor cycle time of 18.5 nanoseconds on Models 200 and 400. The number of cycles per instruction, for any given workload, is a measure of the machine organization efficiency and is relatively independent of the technology. However, different workloads may execute varying mixtures of instructions, yielding a wide range in the average number of cycles per instruction for any given machine. Thus, no single cycles-per-instruction number can be used to describe a processor.

Whereas the cycle time is predominantly a function of technology parameters, and the number of cycles per instruction is dominated by the organizational characteristics of the machine, there is a distinct trade-off to be made between the two. For example, adding more organizational complexity in the machine to improve the number of cycles per instruction requires, in general, the addition of more hardware to the central processor. More hardware implies the potential for increasing the distance between the components to be interconnected and, therefore, an increase in the cycle time. The 3090 central processor combines a sophisticated machine organization with a fast machine cycle time,[1] thereby delivering a substantial improvement in performance compared to an IBM 308X central processor.

The IBM models with multiple central processors deliver more performance for customer applications in a tightly coupled environment. The keys to achieving satisfactory performance from the multiple-processor configurations are an efficient structure to couple more than one processor, and a design which ensures that various buses and other shared resources are in balance for the anticipated load from the central processors.[1]

Input/output subsystem. The input/output subsystem consists of devices attached to the processor via channels. In a typical configuration, several

different types of devices may be attached to the processor, each type satisfying a unique set of requirements. These devices are connected to channels via control units which perform the control and data transfer functions between the devices and the channels. Some examples of input/output devices that may be attached to the processor are local and remote terminals, tape drives, printers, direct access storage devices (DASD), and mass storage subsystems (MSS).

It has become important to store data on devices that have attributes consistent with the frequency of reference.

With the advent of on-line processing of information, it has become important to store data on devices that have attributes consistent with the frequency of reference. Thus, more frequently referenced data are stored on DASD, whereas archival data are typically stored on tape and MSS. Over the last ten years, improvement in processor performance has outpaced the performance improvements in the DASD subsystems. Therefore, the large systems are becoming increasingly dependent upon the performance of DASD subsystems, and require configuration tuning and optimization to realize the full performance potential of the processor.

To ensure a balanced system, two actions are essential: (1) Configure a proper amount of hardware (channels, control units, and devices); and (2) Balance the load reasonably across these hardware resources. As the throughput potential of the processor subsystem is increased, it demands a higher rate at which data requests must be satisfied by the DASD subsystem. This may be achieved by augmenting the DASD configuration or by using faster DASD devices. In the remainder of this section we discuss the key performance attributes of the channel subsystem and the DASD subsystems.

Channel subsystem performance attributes. The channel subsystem capability is determined, from a system viewpoint, by the number of channels, by the data-rate capability of the individual channels and the aggregate channel subsystem, and by the ability to initiate and complete the channel programs at a rate consistent with the peak requirements. Peak requirements are a function of the customer applications and the system services. The 3090 channel subsystem was designed and optimized on the basis of analysis of customer data and the characteristics of control units and devices. For example, the 3090 Model 200 allows up to 48 channels to be configured, of which up to four channels can be operated in a byte-multiplexor mode. Each block-multiplexor channel can operate at up to three megabytes per second in data-streaming mode. The channel configuration options allow for the attachment of the I/O configuration required to realize the full performance potential of a 3090 processor and also provide the capability to satisfy the required channel program rates.

DASD subsystem performance attributes. The DASD subsystem performance plays a very important role in the overall system performance. The DASD subsystem not only accounts for the majority of channels configured on the system to support data access requests, but also, because of its long access time relative to processor speed, accounts for a very large proportion of the time a transaction (or job) is resident in the system. In fact, the DASD component of the response time is the single largest component of the end-user response time, with the possible exception of transmission delays encountered in the communication network. Because response time is the measure of performance for an end user and because it influences end-user satisfaction and productivity, significant effort is expended to optimize the DASD subsystem performance (response time and throughput). In addition, the DASD subsystem response times directly influence the multiprogramming level in the system. Higher multiprogramming levels, in turn, result in a requirement for more processor storage and add to the length of various queues being managed by the operating system, thereby degrading the software efficiency. This phenomenon, of-

ten called the "large-system effect," is discussed later in this paper.

The DASD response time is the sum of the DASD service time and the queueing delay (i.e., the software and hardware wait time prior to finding the device and a path to the device free). The DASD service time is the sum of the following components: seek time, rotational latency, channel and control unit protocol time, search time, data transfer time, and reconnect delay. As a fraction of DASD service time, the components seek, latency, and reconnect delay generally account for 70 – 80 percent of that time. The other service time components—channel/control unit protocol time, search time, and data transfer time—account for the rest and also, collectively, account for the channel/control unit path busy time. The seek and latency times are a function of DASD device characteristics and data reference pattern. The data transfer time, channel and control unit protocol time, and search time are a function of both hardware and software.

Several enhancements have been made to improve the DASD service and response times to satisfy performance requirements.

Reconnect delay represents the component of service time that is due to the path(s) between the DASD device and channel being busy when the DASD device wants to re-establish the connection with the channel in order to search for and transmit the data to the system. If the path is busy, no reconnection is made and, one rotation later, another attempt is made to reconnect. This sequence of events continues until the reconnection is successfully made, with the delay for missed rotations being termed reconnect delay. This delay is a probabilistic function of path utilization and DASD configuration that increases rapidly with path utilization. To keep the reconnect delay at a reasonable level, DASD channel utilizations have traditionally been limited to 20 – 30 percent for interactive environments.

Over the last few years, several enhancements have been made to improve the DASD service, and, therefore, response times to satisfy the performance requirements in the large-system environment. Technological advances have improved the seek time as well as the data transfer rates of DASD devices. Through a combination of hardware and software enhancements, channel and control unit protocol times and search times have also been improved.

The System/370 Extended Architecture (370-XA), in conjunction with MVS/XA and the IBM 3880/3380 systems, addresses the reconnect delay component of the service time.[3] The System/370 architecture, prior to the introduction of 370-XA, required that all operations related to an I/O request be conducted on the same path on which an I/O operation had been initiated. Therefore, any reconnection had to be made on the same path, even though there might exist other available paths between the DASD and the host system. The 370-XA architecture permits the reconnection to be made on any available path between the DASD and the host system. With this architectural enhancement, a reconnect attempt fails only if all paths to a DASD device are busy, the probability of which is significantly reduced. This architectural enhancement represents a very fundamental advance in the DASD subsystem performance by reducing the reconnect delay. From a performance viewpoint, this capability manifests itself as an improvement in DASD service time (and response time) for a specified workload or as the ability to support higher throughput for a specified response time objective. At the channel, the latter is reflected as the ability to achieve higher channel utilizations without degrading performance.

Further enhancements to reduce the reconnect delay have been implemented via device-level switching. This feature, available on newer models of IBM 3880/3380, makes the dynamic reconnection capability operate on a finer granularity than was previously available. Once again, the result is either more throughput (accesses per second) or improved response time from the DASD configuration.

We now consider the performance of the cached DASD subsystems.[4] During the design phase of the 3090, analytical evaluations showed that additional improvements in the I/O subsystem performance would be of considerable importance for a processor in that performance range. With the continued improvements in the performance, cost, and reliability of semiconductor technology, a cached DASD subsystem (e.g., the 3880 Models 13 and 23 used for data base and program libraries) offers a much-needed improvement in DASD subsystem service times. A cached DASD subsystem consists of a control unit with a large semiconductor buffer, managed as a cache by the control unit so as to buffer the more frequently used data residing on the DASD devices attached to the control unit. Data references can be satisfied at electronic speeds without electromechanical delays, if the required data are resident in the cache, with the misses being serviced by the DASD. Write operations are handled by recording the updated record on the magnetic media, in addition to updating the cache, if required, to ensure data integrity in the event of losing the contents of the cache. Clearly, the average service (and response) time of the cached DASD subsystem is strongly related to application-dependent parameters such as the hit ratio, read-to-write ratio, and record sizes. In general, however, the cached DASD subsystem offers a significant reduction in service and response times when configured for appropriate applications, and should play a major role in realizing the full performance potential of the 3090.

Storage. An application's view of storage is its virtual address range; the programs and data referenced by the application reside in virtual pages within the address range. System/370 architecture provides a virtual address range from zero to sixteen megabytes, which is composed of 4096 virtual pages with each page 4096 bytes in size. The architectural extensions of 370-XA increase the virtual address range to two gigabytes (524 288 virtual pages). An application generally uses far less than the up-to-two gigabytes of virtual storage available to it. A system may support hundreds of applications simultaneously, and requires storage to contain all active virtual pages.

Conceptually, there are two classes of storage configured to a system—processor storage and auxiliary storage—in which virtual pages may reside. The distinguishing characteristic of the two classes is the manner in which each class of storage is accessed. *Processor storage* is accessed synchronously with the processor. That is, the processor waits while a data item is retrieved from this class of storage. *Auxiliary storage* is accessed in an asynchronous manner. That is, an I/O is scheduled to retrieve a data item from this class of storage, and the processor is free to execute some other task. In either case, the application waits while its requested data item is retrieved. Since processor storage is integrated with the processor, retrieval time from it is measured in microseconds or less. Auxiliary storage, usually defined on a rotating device attached to the processor through a channel and control unit, is characterized by retrieval times measured in milliseconds. This speed difference gives processor storage a tremendous performance advantage over auxiliary storage. However, the cost of configuring enough processor storage to contain all applications' virtual pages is prohibitive. A system must have a balance between the two classes of storage, so that the most frequently referenced virtual pages reside in processor storage and the remainder reside in auxiliary storage.

The 3090 provides two categories of processor storage: central storage (commonly called real storage) and expanded storage. Central storage is connected to the high-speed buffer (cache) of a processor where instructions and data must reside prior to execution. Expanded storage is connected to central storage. Both are accessed in a synchronous manner. When an instruction or data item is referenced and does not currently reside in the high-speed buffer, the processor waits for it to be retrieved. If the item is in central storage, it is moved directly into the high-speed buffer; if the item is in expanded storage, it is first moved into central storage and then into the high-speed buffer. Transfers between the high-speed buffer and central storage occur in 128-byte groups under hardware control. The operating system controls transfers of 4096-byte pages between central storage and expanded storage. Central storage is architecturally limited to two gigabytes, whereas expanded storage is expandable to sixteen terabytes.

Processor storage is managed by the operating system in a least-recently-used (LRU) manner. That is, the virtual pages that go unreferenced for the longest time are removed from processor storage to make room for newly referenced pages. With the 3090, central storage contains the most fre-

quently referenced pages; expanded storage contains the moderately referenced pages; and auxiliary storage holds everything else. An illustration of the virtual-page-reference activity of a system versus where the page is located is presented in Figure 2. The shape of the curve is a function of the page reference patterns of the applications running on the system and the LRU management of processor storage. A well-balanced system will find the steep area of reference activity contained in processor storage and the flat area satisfied by auxiliary storage. If the processor storage is too small to contain the frequently referenced pages, a classic thrashing situation is likely to occur. The major value of processor storage is its effect on response time, but it can also improve the capacity of a system to process useful work. On a storage-imbalanced system, response time can be severely affected by numerous paging I/O delays (retrieving pages from auxiliary storage). Additionally, the processor may be idled frequently while all applications wait for paging I/O to complete. With sufficient processor storage, these delays can be reduced to a minimum, improving both the response time and the ability of the system to process work.

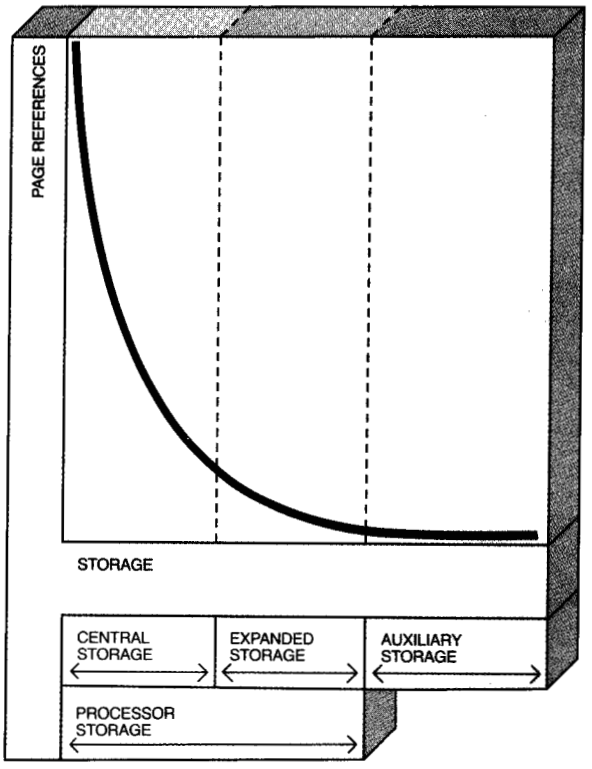
The amount of processor storage required for a balanced system is very workload-dependent. The response time goals, types and sizes of the applications, transaction rates, and number of users must be understood in order to derive a reasonable estimate.[5] The processor storage options on the 3090 should allow any workload to achieve a proper balance. In addition, the large storage options present opportunities to reduce I/O delays in areas other than paging by using processor storage to buffer programs (e.g., IMS preloading and virtual fetch and CICS dynamic storage area) and data (e.g., larger data base buffer pools).

Software

The responsibility of the software is to combine the components of a balanced system in a synergistic manner. The operating systems MVS/XA, VM/SP-HPO, and VM/XA System Facility have been designed to efficiently manage the processor, I/O, and storage resources, allowing the full capacity of a 3090 system to be used in a productive manner.

MVS/XA. MVS/XA[6] provides the system software base on which applications running under MVS

Figure 2 System page reference activity



can continue to grow. Through the exploitation of the 370-XA architecture, MVS/XA provides significant enhancements over MVS/370 in all major areas of resource management.[7]

In the area of processor management, the major goals of MVS/XA are to eliminate *large-system effects* and to provide a structure which can allow for a high level of parallelism in a multiprocessing system. A large-system effect is typically an internal MVS function which operates efficiently on existing systems but degrades on new, larger systems and results in increased processor time to perform the identical function. Previously, these functions were often designed to manage a pool of resources (e.g., devices, storage frames, or user address spaces) of a certain size. If the size of a pool grew substantially on a larger system, the efficiency of the function might suffer. Many components of MVS, such as the real storage manager, the virtual storage manager, the locking/serialization managers, and the I/O supervisor, were completely redesigned for MVS/XA, focusing on the removal of

large-system effects and supporting a high level of parallel processing.

One of the most significant goals of MVS/XA is to address the storage constraints inherent in the 24-bit addressing scheme of System/370. As applications grow, 16 megabytes of virtual addressability becomes a potential limit to fully utilizing the capacity of a large system. MVS/XA supports 31-bit virtual storage addresses, thereby providing two gigabytes of addressability. This enhancement provides the base upon which applications and subsystems may continue to expand. MVS/XA supports 31-bit real storage addresses, meaning that systems with real storage requirements up to two gigabytes can be accommodated. Note that MVS/370 supports up to 64 megabytes of real storage, but management efficiency suffers beyond a 32-megabyte configuration. MVS/XA is the only level of MVS to support the 3090 expanded storage option.

The auxiliary storage manager has improved algorithms for managing the local paging devices.

MVS/XA exploits the new I/O architecture available with 370-XA. The number of channel paths supported has grown from 32 in MVS/370 to 256 with MVS/XA. Likewise, the number of devices which can be defined to a single system has been raised from approximately 1200 to 4000. The dynamic pathing supported by MVS/XA provides a structure in which the pathing selections are done by the channel subsystem. Effectively, this encourages an I/O to be started and completed via the most available paths (channel and control unit) to the device. Dynamic pathing can provide improvements in the processor time and especially in the response time to perform an I/O operation.

MVS/XA provides many other performance enhancements. The *linklist lookaside* function allows an in-storage directory search for modules in the system linklist, thereby reducing the I/O overhead involved in loading a program. The auxiliary storage manager has improved algorithms for managing the local paging devices in a system's paging configuration, thus leading to smaller configurations and less time spent tuning. I/O measurement and reporting have been enhanced for channels, control units, and devices.

VM. VM support of the 3090 is provided with the VM/SP High-Performance Option (HPO) and the VM/XA System Facility.

HPO provides the system software base on which System/370 VM applications can be expanded to new or larger IBM processors. Through VM/SP modifications and extensions, HPO provides significant improvements, as measured in throughput and response time on large processors.[8] VM/XA System Facility provides the software base for the use of the 3090 for System/370 Extended Architecture (370-XA) migration and guest operating system support.[9]

Large-system effects have been handled for VM through the use of hardware assists, microcode, and software extensions. VM system functions such as auxiliary storage management have been rewritten to utilize processor storage greater than 16 megabytes and to add block paging and swapping. Software changes have been made to improve dyadic performance through restructuring of control blocks and their alignment. Other changes are the creation of processor local queues to improve the efficiency of the high-speed storage buffer, and improvements in the area of interprocessor communications. Processor functions such as segment protection and microcode assists such as the preferred machine assist have been utilized to reduce system overhead and improve performance.

The balancing of system performance for VM/SP HPO on the 3090 requires access to more than 32 channels on the dyadic (Model 200) in 370 mode. This implies a capability to support more than 16 channels per channel set. The 3090 provides 32 channels per channel set, thereby allowing access to all 48 channels of the Model 200. These additional channels provide the added I/O paths to

balance that portion of the system with the improved performance of the 3090.

HPO is required to support the System/370 VM balanced growth opportunities of the 3090. VM/XA System Facility, which is the key VM product for 370-XA, provides for the guest operation of MVS/XA and other operating systems on the 3090. Through the use of Start Interpretive Execution (SIE) mode and the SIE assist, VM/XA SF provides efficient support for the preferred guest MVS/XA operating system as well as extensive system programmer tools for the maintenance and support of guest systems.

Performance

The value of a computing system is determined by the speed at which it performs the work of a user's installation. The 3090 may be used to satisfy a wide range of commercial and scientific applications, including DB/DC, interactive, and batch. With this flexibility comes the likelihood of varying performance expectations, depending on the workload to be processed.

As an example of comparison, the 3090 Model 200 was measured while it was processing representative workloads from several major areas of application. Since the measurements were conducted in a controlled environment, care should be taken when applying the results to other operating environments where significant variations may be present.

Metrics. Many units of measure have been used to assess the performance capabilities of computing systems. Three popular metrics, instruction execution rate, throughput, and response time, are discussed here.

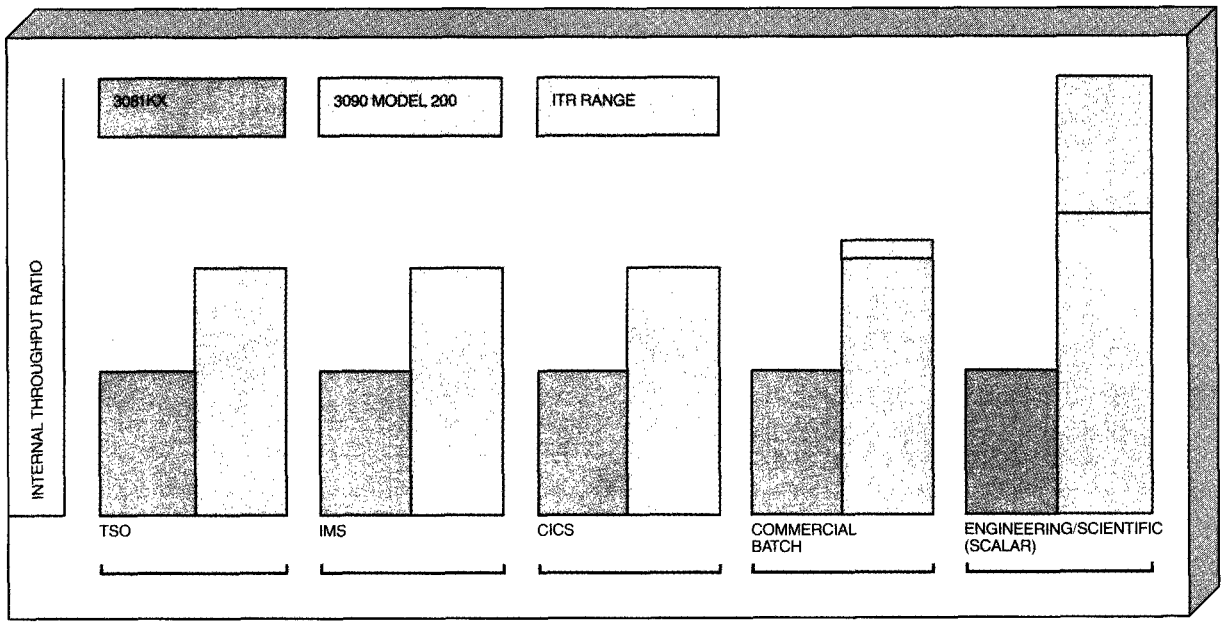
Instruction execution rate. The instruction execution rate (IER) is measured while the processor is busy. It does not reflect the number of instructions executed to do a particular amount or kind of work, nor is there any indication as to whether the work was productive.

Several major factors influence the IER of a processor. The type of workload being processed can produce a wide variation in the IER of a processor. For example, a scientific workload may run at twice the IER of a commercial workload. The instruction set architecture and level of microcode implementation can affect the IER comparison of two processors running the identical workload. For example, if a function requiring a large number of instructions on one processor is implemented as a single instruction on another, the IERs of the two processors may vary, depending on how frequently the function is executed. Given the potential for variation, whether measuring the same processor with different workloads or different processors with the same workload, instruction execution rate is not a reliable performance metric.

Internal throughput indicates the potential capacity of the processor to perform work.

Throughput. The throughput of a system can be defined as the useful work it can complete per unit time. Transactions per second and jobs per hour are common units used to express throughput. Throughput is generally calculated as an external measure of performance, the count of transactions or jobs completed divided by the elapsed time over which the count was taken. Thus, external throughput indicates what the system actually processed. However, this raises the question of whether external throughput reflects the potential *capacity* of the processor to perform work. Here the metric is not necessarily valid because, by the nature of the workload, a processor is likely to be idle for periods of time when there is no more work to be processed or when there are outboard influences, such as I/O or operator delays. If this idle time were to be removed from the denominator of the throughput equation, the resulting *internal* throughput would reflect the actual capacity of the processor, because it would appear to be busy 100 percent of the time. Thus there are two measures of throughput: (1) External throughput measures the actual rate at which work is performed by a system; (2) Internal throughput measures the rate at which a processor is capable of executing work.

Figure 3 Internal throughput ratio with MVS/SP2.1.3



The 3090 can be analyzed both as a processor and, by including its channel and storage options, as a system. To illustrate its performance capabilities, both internal and external throughput are used as appropriate.

Response time. Response time is of the utmost importance to the end user. To a large extent, the productivity of the end user is governed by the level of interactive responsiveness of the computing system.[10] In an analysis of the components of response time, I/O time tends to be the dominant component, often accounting for 90 percent or more of the total. Because the processor accounts for only 10 percent or less, response time is not an appropriate measure of processor performance.

However, response time can be useful for comparing systems. One system may offer additional I/O capabilities, whereas another may offer more processor storage to reduce paging I/O delays. In these cases, response-time comparisons can be meaningful, provided that each system has been tuned to make best use of the resources it supports.

3090 performance with MVS/XA

Processor capacity. The capacity of the 3090 to execute work can be illustrated, for example, by comparing one of its models with its predecessor in the 308X family that has a like number of processing engines. In this example, we compare a 3090 Model 200 with a 3081KX. Generally speaking, other model-to-model comparisons exhibit similar relationships. Representative workloads from five major application areas were measured. Ratios of internal throughput rates of the 3090 Model 200 to the 3081KX for each category are summarized in Figure 3.

Interactive and DB/DC applications, TSO, IMS, and CICS were compared using measurements made while each processor was approximately 70 percent busy. This represents reasonably heavy utilization of a system that is dedicated to a response-oriented workload. Each of the workloads showed a ratio of approximately 1.7 between the 3090 Model 200 and the 3081KX. TSO user loads varied from 520 users on the 3090 Model 200 to 310 users on the 3081KX. Simulated as locally attached terminals by an internal driver, the TSO users executed program development scripts that included full-screen editing, compilations, and interactive testing. IMS/VS Version 1.3 was measured at 40 transactions per second on the 3090 Model 200 and 23 transactions per second on the 3081KX. CICS/VS Version 1.6.1 was measured at 85 transactions per second on the 3090 Model 200 and 50 transactions

per second on the 3081KX. The IMS and CICS loads were generated by an internal driver simulating users involved with order entry, inventory tracking and control, banking, and reservations. Although the workload scripts were similar, IMS and CICS did not count transactions in the same manner, so the transaction rates of the two subsystems cannot be compared.

Several batch workloads were measured, providing a range of internal throughput ratios. Two commercial batch job streams consisting of a variety of COBOL, SORT, PL/I, and utility applications each yielded 3090 Model 200 to 3081KX ratios of approximately 1.8. A workload characteristic of application program development using FORTRAN showed a ratio of 1.9 between the 3090 Model 200 and the 3081KX. Comparisons of the 3090 Model 200 to the 3081KX running individual engineering/scientific scalar application programs found ratios from 2.1 to 3.1. This variation in the internal throughput ratio for engineering/scientific programs resulted largely from the superior floating-point operations implementation of the 3090 compared to the 308X. The percentage of floating-point operations in the workload correlated with the internal throughput ratio, the higher percentage of floating-point yielding the higher ratios.

Expanded storage. The 3090 introduces a new type of processor storage, called *expanded storage*, as described in an earlier section. MVS/XA manages expanded storage as an extension of central storage. The management algorithms are designed so that an increment of expanded storage reduces paging I/O by an amount similar to the reduction obtained if an equal increment of real storage were added to a system.

The major benefits of adding either category of processor storage to a system are reduced paging I/O (allowing a reduced paging configuration) and improved response time. Measurements with a combined TSO and commercial batch workload were performed to illustrate these benefits with expanded storage. The following three system configurations were used: (1) 3090 Model 200, with 64 megabytes of central storage and twenty 3380 paging devices; (2) 3090 Model 200, with 128 megabytes of processor storage (i.e., 64 megabytes of central and 64 megabytes of expanded storage) and twenty 3380 paging devices; (3) 3090 Model 200 with 128 megabytes of processor storage (i.e., 64 megabytes of central and 64 megabytes of expanded storage), and six 3380 paging devices. Figure 4 shows the relationships among the total paging rates to auxiliary storage devices, the TSO user load, and the system configurations. Both of the 128-megabyte configurations provided significant reductions (from 60 to 70 percent) in paging at all measured TSO load points.

Figure 4 Total paging rate to auxiliary storage

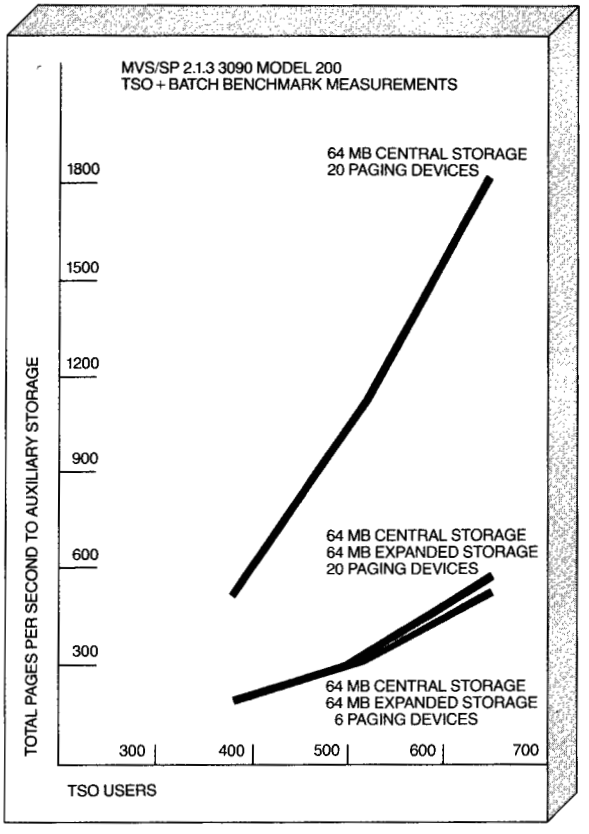


Figure 5 illustrates the effects on the average response time of trivial TSO transactions, which account for about 90 percent of the total transactions in the workload. The addition of 64 megabytes of expanded storage provided excellent response time across all TSO load points. In addition, the reduced paging rate allowed a two-thirds reduction in auxiliary storage paging configuration with little effect on response time.

We now discuss the benefits of expanded storage compared with those of real storage. We compared

Figure 5 Trivial TSO response time

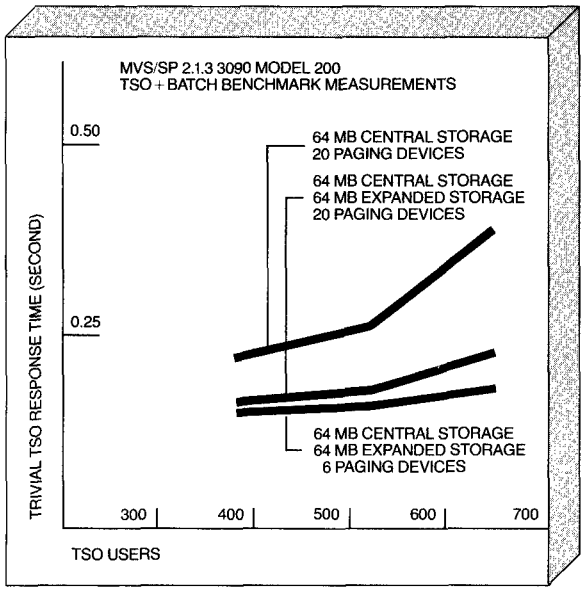
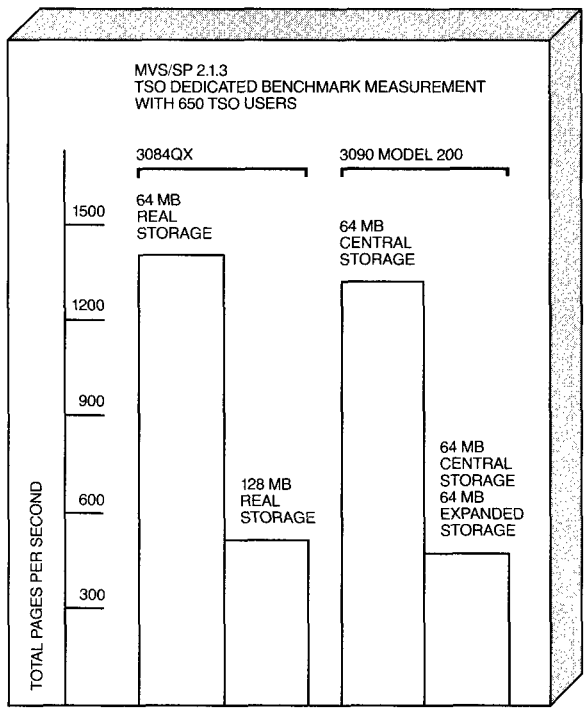


Figure 6 Total paging rate to auxiliary storage



the addition of equal increments of each type of storage to processors of similar capacity. A 3084QX provides approximately the same processing capacity as a 3090 Model 200 and can be upgraded from 64 megabytes of real storage to 128 megabytes of real storage. A 3090 Model 200 is available with 64 megabytes of central storage and up to 256 megabytes of expanded storage. Therefore, we compared the benefits of adding 64 megabytes of real storage to a 3084QX to those of adding 64 megabytes of expanded storage to a 3090 Model 200. Measurements were performed using a dedicated TSO workload. The following four system configurations were used: (1) 3084QX with 64 megabytes of real storage; (2) 3084QX with 128 megabytes of real storage; (3) 3090 Model 200 with 64 megabytes of central storage; (4) 3090 Model 200 with 64 megabytes of central storage and 64 megabytes of expanded storage. The I/O and paging configurations of each system were similar. As may be seen from Figure 6 (paging to auxiliary storage) and Figure 7 (trivial TSO response time), the benefits of the additional storage on the 3090 Model 200 and the 3084QX were very similar.

3090 performance with VM

Interactive performance. The interactive performance previously described is a system measure of performance and is a function of the processor, storage, and I/O of the system, as well as the type of work being performed. For HPO, the interactive performance is best evaluated using trivial response time, which is the response time for those transactions that can be completely serviced in a single time slice.

Interactive measurements were made using a laboratory workload that is representative of commercial customer interactive environments. This commercial CMS workload utilizes simulated 3270 full-screen applications to represent local users and is based on VM/SP3 CMS using IBM BASIC, assembly language, PL/I, DCF, COBOL, APL, FORTRAN, and VMAP. Each of the CMS virtual machines requires two megabytes of virtual address space and has more than 3000 files in its disk search order.

The balanced growth capability of the 3090 is well demonstrated with the HPO measurements where the gain of each new facility of the 3090 can be isolated. In the laboratory evaluation of HPO Release 3.6, trivial response time was measured on the 3090 Model 200 and the 3081KX. Both pro-

cessors had 64 megabytes of real storage. First, measurements were made on the 3081KX with the system tuned to achieve the maximum performance possible with the laboratory workload. The 3090 Model 200 was measured with 24 and with 32 channels and with expanded storage. The general results of these measurements are shown in Figure 8. Because of the low portion of response time attributable to the processor, the impact of the faster 3090 Model 200 processor is minimized in this comparison, and the trivial response time curve for the 3090 Model 200 without additional channels and expanded storage is equivalent to that of the 3081KX.

At a workload level saturating the capacity of the 3081KX (i.e., with 300 users and the laboratory workload) and with the I/O configuration enlarged and tuned to take advantage of the 3090's extra capability, there is a 70 percent reduction in trivial response time over that of the 3081KX with the same number of users. This improvement is due to a reduction in the amount of I/O wait time that

Figure 7 Trivial TSO response time

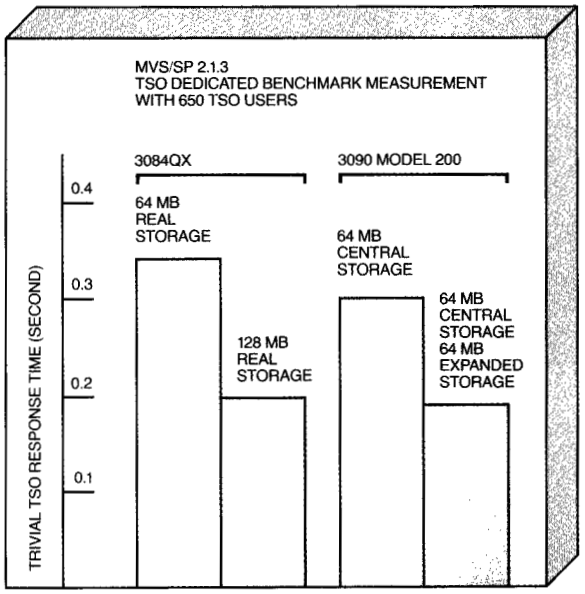
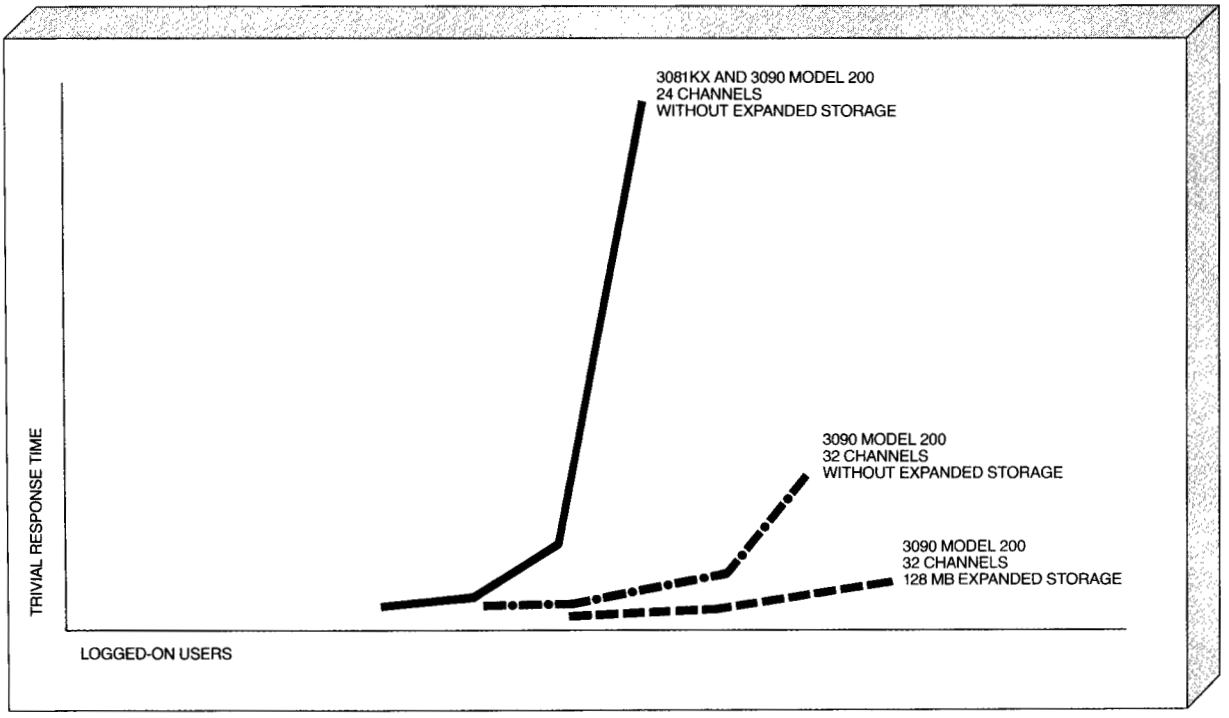


Figure 8 VM/HPO Release 3.6 interactive performance



is made possible by the additional channel paths of the 3090.

Figure 8 also shows the trivial response time of the 3090 Model 200 with all features exploited. The addition of 128 megabytes of expanded storage provides an additional 10 percent reduction in trivial response, yielding a total improvement of 80 percent over the 3081KX with 300 users. Figure 8 also shows that the fully configured 3090 Model 200 provides an 80 percent increase over the 3081KX in the number of users supported.

Engineering/scientific scalar batch performance. The capacity of the 3090 processor to execute work under HPO can be shown in an engineering/scientific environment. Measurements were made to compare the external throughput of a 3090 Model 200 to a 3081KX. Each system was configured with 64 megabytes of real storage. Several scalar engineering/scientific FORTRAN programs obtained from IBM and customer laboratories were measured. For these sample workloads, the 3090 Model 200 shows an improvement in external throughput ratio over the 3081KX similar to the gain in the MVS/XA environment. It is important to note that throughput improvement for other engineering/scientific work depends on the instruction mix and the amount of other work present on the VM system.

Concluding remarks

The 3090, which is the largest IBM system to date, offers a cost-effective solution to customer growth requirements. Its channel configuration options and—with the introduction of expanded storage—its processor storage options provide for a balanced system to enable customers to realize the full performance potential of the 3090 in their environments. Improvements in the I/O subsystem performance have been made through architectural enhancements and cached DASD subsystem to achieve an overall balance at the system level. The MVS/XA, VM/SP-HPO, and VM/XA System Facility are designed to realize the performance potential of the 3090. In laboratory benchmark environments of comparable 3090 and 308X models, the 3090 has provided an internal throughput ratio of from 1.7 to 1.9 for commercial environments and up to three times for specific scientific applications.

Acknowledgments

Numerous individuals have contributed to the evaluation and optimization of the performance of the 3090, thus ensuring a balanced system offering for satisfying customer requirements. Because it would be impractical to acknowledge their contributions individually, the authors would like to recognize them as teams. The System Performance Analysis organization identified the customer performance requirements and the 3090 configuration offering required to maintain a balanced system. The Processor Performance Analysis group participated in the design analysis and optimization of the 3090 performance. The Performance Evaluation organizations in Poughkeepsie, Kingston, and the Washington Systems Center conducted the product measurements and evaluation to quantify the product performance attributes, including specialized experiments, as required, for the MVS and VM operating systems.

Cited references

1. S. G. Tucker, "The IBM 3090 system: An overview," *IBM Systems Journal* **25**, No. 1, 4–19 (1986, this issue).
2. R. J. Wicks, *Balanced Systems and Capacity Planning*, GG22-9299, IBM Corporation; available through IBM branch offices (November 1982).
3. U. Pimiskern, *MVS/XA I/O Performance Considerations*, GG22-9346, IBM Corporation; available through IBM branch offices (December 1983).
4. R. J. Wicks, *DASD Expectations: The 3380, 3880-23 and MVS/XA*, GG22-9363, IBM Corporation; available through IBM branch offices (July 1985).
5. S. N. Allen and G. M. King, *MVS/XA Processor Storage Estimates and Performance*, GG22-9397, IBM Corporation; available through IBM branch offices (April 1985).
6. *MVS/Extended Architecture Overview*, GC28-1348, IBM Corporation; available through IBM branch offices.
7. E. I. Cohen, "MVS/XA performance considerations," *Proceedings of SHARE 64*, Los Angeles, CA, February 25–March 1, 1985.
8. *Virtual Machine System Product Introduction*, GC19-6200, IBM Corporation; available through IBM branch offices.
9. *Virtual Machine/Extended Architecture System Facility, General Information*, GC19-6213, IBM Corporation; available through IBM branch offices.
10. A. J. Thadhani, "Interactive user productivity," *IBM Systems Journal* **20**, No. 4, 407–423 (1981).

Yogendra Singh *IBM Data Systems Division, P.O. Box 390, Poughkeepsie, New York 12602.* Dr. Singh is Program Manager, Future Processor Development. He joined IBM in 1972 as a staff engineer with the Systems Development Division. Among the positions he has held, Dr. Singh was manager of large-system performance evaluation in the Data Systems Division at Poughkeepsie. He received an Outstanding Innovation Award in 1982 for his work on large-system performance. Dr. Singh

received his B.Tech. in electrical engineering from the Indian Institute of Technology, Kanpur, in 1968 and his M.S. and Ph.D., both in electrical engineering, from the University of Illinois at Urbana-Champaign in 1970 and 1972, respectively. He is a member of the IEEE.

Gary M. King *IBM Data Systems Division, P.O. Box 390, Poughkeepsie, New York 12602.* Mr. King is a senior programmer in the MVS Design and Performance Analysis Department. He joined IBM in 1974 as an associate programmer with the Systems Development Division. Mr. King has been involved in the design and evaluation of the MVS resource managers, especially in the storage management area. He has received Outstanding Technical Achievement Awards for the development of a technique to study virtual storage page reference patterns (1982) and for the design of expanded storage management (1985). Mr. King received a B.S. in mathematics from the State University of New York at Albany in 1972 and an M.S. in computer science from the Pennsylvania State University in 1974.

James W. Anderson, Jr. *IBM Data Systems Division, Neighborhood Road, Kingston, New York 12401.* Mr. Anderson received his B.S. in electrical engineering from Michigan State University in 1965. He joined IBM as a programmer in the Poughkeepsie Product Test Laboratory, testing operating systems from OS/360 Release 1 through MVS. He transferred to Kingston in 1973 as Manager of Programming Development Information Services, supplying VM and MVS services to Kingston programmers. In 1983, Mr. Anderson joined the high-end VM development group in Kingston. He has served as VM/SP HPO design manager and HPO design and development manager; and he is currently responsible for the performance evaluation and measurement of VM/SP HPO and the VM/XA System Facility.

Reprint Order No. G321-5259.