Cache-DASD storage design for improving system performance

by C. P. Grossman

This paper discusses three examples of a cache-DASD storage design. Precursors and developments leading up to the IBM 3880 Storage Control Subsystems are presented. The development of storage hierarchies is discussed, and the role of cache control units in the storage hierarchy is reviewed. Design and implementations are presented. Other topics discussed are cache management, performance of the subsystem, and experience using the subsystem. It is shown that a cache as a high-speed intermediary between the processor and DASD is a major and effective step toward matching processor speed and DASD speed.

From the infancy of computers and data processing to the present, there has been a continuing effort to improve the speed of the input/output (I/O) devices attached to the central processing unit (CPU). The card reader and punch that were used for input and output in the ENIAC, which was developed for ballistics calculations during World War II and was the first stored-program computer, were much slower than the vacuum tubes and relays that constituted the ENIAC's central processing unit.[1] As computers developed over the years, input/output technology also changed—from punched cards, to paper tape, to magnetic tape, to the first direct access storage device (DASD) with movable heads (the IBM 350 RAMAC),[2] and on to the current class of disk drives as represented by the IBM 3380. During the same period, processor technology has advanced at an even faster pace, so that the gap between I/O device performance and processor performance has widened.

Throughout this evolutionary process, three conflicting factors have been important: higher performance, larger capacity, and lower cost. Many different technologies have been used to balance these three design factors in the development of storage devices.

The precursor of today's direct access storage device (DASD) was the magnetic storage drum device,[3] developed for commercial use in the early 1950s. In this implementation, a rotating cylinder with a magnetic coating had data stored on tracks encircling the cylinder. Each of these tracks had a magnetic read/write head permanently fixed in place to read and write data, as illustrated schematically in Figure 1. Because of the high cost of the read/write heads, magnetic disk devices with movable heads were developed. The IBM 350 RAMAC, introduced in 1956, used a single read/write head that could move from disk to disk or across a disk surface. This design was further refined to disk drives that had a movable head for each disk surface. Examples include the IBM 3330 removable disk drive and the fixed-media 3380.[4] The movable-head design is also illustrated in Figure 1.

The evolution of disk devices has shown steady improvements in performance, increases in storage capacity, and decreases in the cost of storage. However, because of the mechanical limits of DASD, many other techniques have been used to provide greater speed for at least the most important data used by a computer.[5]

One of the early designs is exemplified by the IBM 2305. This device has a number of disks, each with concentric tracks for storing data. Like the earlier drum, it has a fixed read/write head over each track on each disk surface. It is, however, more expensive than a movable-head design, though it does eliminate the mechanical motion of the head known as *seeking*. For many years, this was the device of choice for improving the storage and retrieval performance of a system's most critical data. The fixed-head configuration is illustrated in Figure 1.

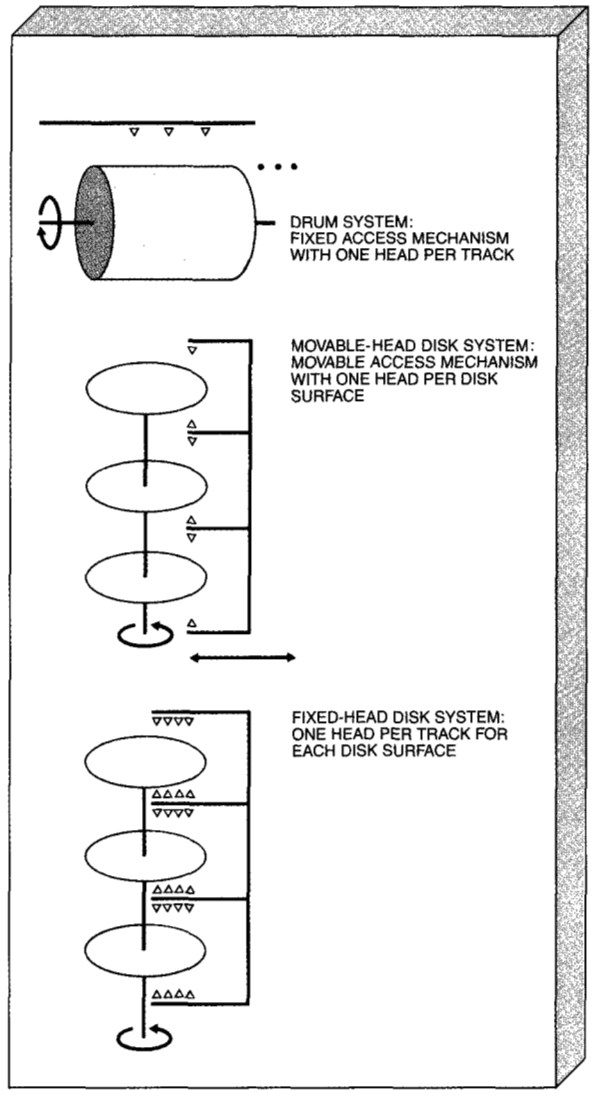
A variation of this design is to install a small number of fixed heads on a conventional disk device, as was done with the IBM 3350. The user was required to place high-performance data under the fixed heads. Although this design gave performance improvement, storage space was limited, the user had to manage that space, and there was a possibility of interference from other, less important activity on the device.

As large-scale integrated circuits became more reliable and more densely packed, they made solid state electronic storage commercially feasible. Electronic storage, like electronic main storage, was volatile. That is, a loss of power would result in a loss of data. These devices have random access to data and they read and write data at electronic speed, without the electromechanical motion associated with disk devices. Because of the volatile nature of this electronic storage, unless there is a battery backup, these devices are appropriate for read-only data, for data that can be easily recreated and restored, or for data such as pages in a virtual storage operating system, which are not carried across system initial program loads (IPL).

A relatively new approach to matching DASD performance with that of the CPU uses the concept of a data storage hierarchy to improve disk performance. Although this is new to DASD, the concept of the hierarchy has been used for many years in other components of a computer system. This is the idea behind the cache memory, which was first used by IBM in the System 360/Model 85.[6]

The processor cache is a high-speed, high-cost storage unit used as an intermediary between the central processing unit and main storage. Just as I/O devices are slower than the processor, storage fetch cycles are also slower than CPU cycles. Therefore, the use of the high-speed cache as a buffer allows the CPU to fetch and process data at a much greater speed, thereby more closely matching storage speed with that of the CPU.

Figure 1 Three different direct access storage systems



For disk drives, the principle of a cache has been implemented by IBM in the 3880 Models 13 and 23 Cache Storage Controls.[7] This design can be considered a selection of the best features of disk storage and solid state storage devices. The data are permanently resident on the disk (for data integrity), and a copy of the more active data can be stored in the cache for fast access. Microprocessors within the control unit monitor the use of data and help ensure that only frequently referenced, active data remain in the cache.

This paper explores the notion of data hierarchies and their roles in the design process for the IBM 3880 cache control units. Also presented are the implementation of the 3880 cache and cache management. Performance is discussed in terms of measures of cache effectiveness, characteristics of "cache-friendly" data, and modeled as well as actual performance results. Performance evaluations demonstrate the effectiveness of the initial cache design in improving system performance.

A key requirement for the best price/performance is balancing cost, capacity, and performance.

Hierarchy of data storage

The data available to or used by a processor can be categorized by their frequency of use and the speed of access required. Looking at data in this way suggests a hierarchy of storage media in which the elements may differ in cost, capacity, and speed of access.[3] A key requirement for providing the best price/performance characteristics to a system is that of balancing three factors: cost, capacity, and performance.

The most effective way to balance these factors is to provide a mechanism for data to flow from one storage medium to another, as their usage characteristics change. An example is the implementation of a high-speed cache as a buffer between the central processor itself and its main storage.[3,8] The success of this technique depends on two characteristics of data used by a CPU: *locality of reference* and *data reuse*. Locality of reference means that if a given piece of information is used, there is a high probability that a nearby piece of information will be used very shortly thereafter. A least recently used algorithm (LRU) is employed to ensure that only blocks of data that have been used recently are kept in the cache. Blocks that are not referenced frequently are overlaid by newly requested data. This technique ensures that most of the time the data needed by the CPU are in the high-speed cache. Thus it is possible to use an expensive resource—the cache—very efficiently. The CPU's main storage is much larger, slower, and less costly than the cache. By using the cache for most accesses to storage, however, a request for data is satisfied at the speed of the cache, which more nearly matches the speed of the CPU. Thus, use of the cache facilitates optimum use of the more costly component, the CPU.

This hierarchic approach to storage is implemented in a number of other areas of large systems today. Some, like the cache just described, are implemented solely within the hardware. Others, such as paging in a virtual operating system, are implemented by a combination of hardware and software.[9,10] Still others, such as data migration programs, are strictly software products. Clearly this notion of automatic system-controlled movement of data from one class of storage device to another will be expanded and utilized more fully. One characteristic these systems share is the ability to maximize the value of an expensive resource by moving data that are no longer active off that medium and onto a less expensive one.

Design and operation of the IBM 3880 Models 13 and 23

Locality of reference, discussed earlier with regard to the high-speed cache, is also a well-known characteristic of other classes of storage, the best known of which is virtual storage.[11] Virtual storage makes use of the fact that a relatively small portion of the address space required by a program is actually frequently used. This address space subset is called a program's *working set*. The larger portion of the program not frequently used is temporarily stored on a disk device until needed. Hardware in the CPU (for *dynamic address translation*) and software in the operating system manage the storage and perform the virtual-to-real address translation.

Locality of reference is also used in the IBM 3850 Mass Storage Subsystem (MSS),[4] where a small number of disk drives are made to appear to be a much larger number of *virtual volumes* by an address translation facility analogous to that used in virtual storage. Again, frequently used data tend to gravitate to the real disk drives, and less frequently used data are destaged to tape cartridges stored within the MSS.

The DASD cache control unit is the most recent development in the history of implementation of the

concept of caching data and mapping a large, less expensive, slower storage resource into a smaller, more expensive, faster one.

A DASD control unit is a vital component of the DASD subsystem.[4,9] The control unit is a highly specialized microprocessor attached to the channels of the processor as well as the disk devices. (A channel is a specialized I/O processor that takes the function of device communication from the CPU and allows an architecture capable of driving the processor to a higher level of utilization.) The DASD control unit communicates with the processor by means of the channels and manages the operation of the disk drives. The DASD control unit receives requests for read or write activity from the channel, translates them into the proper sets of orders for the disk device, sees that the orders are executed, performs error recovery, and reports any unusual conditions back to the channel, and hence to the processor.

Earlier DASD control units consisted of one microprocessor that could control up to 32 disk devices (as in the IBM 3830). The current IBM DASD control unit (i.e., the 3880) packages two such microprocessors within one machine. Each microprocessor, called a *storage director*, can control its own set of DASD totally independently of the other microprocessor.

With the development of *cache control units*, such as the IBM 3880 Model 23, the usual functions of a control unit have been augmented by the significant new function of support and management of a cache storage between the disk device and the channel. The function and benefit of this cache are entirely analogous to those of the high-speed cache used in very large processors.

From experience gained working with the design of storage hierarchies, it was clear that there was a significant performance benefit to be gained by applying caching principles to a DASD subsystem. To take this step, the following key design objectives had to be met:

- The reliability, availability, and serviceability of the DASD subsystem would be maintained at a very high level through use of new technology.
- Performance approaching that of solid state storage would be achieved, without the risk of data loss inherent in volatile storage.
- The disk *image* of the data would be maintained. That is, the data would appear to the processor to be in DASD format.
- Customer investment in application programming would be protected. Application programs written to use the standard IBM access methods and without timing dependencies on a specific disk device type would run without modification.
- Operational changes within the data center would be held to a minimum.

The 3880 Model 23 has a number of features designed to improve the availability and reliability of the control unit. Improved error detecting and checking circuitry and microcode allow the Model 23 to correct most hardware triple-bit errors and all double-bit errors. In most cases, if one storage director in a storage director pair fails, the remaining storage director continues using the cache. In addition, after a directory error condition that terminates caching, the Model 23 automatically reconfigures the directory of cache contents and brings the cache storage back on line without operator intervention. The hardware components of the control unit have been designed so that most repairs can be completed without disabling the entire subsystem. Full caching function, or fastest access to data, may not be possible, but the processor is still able to access data through one of the storage directors.

The cache management algorithms were designed to provide the best overall performance in typical data processing installations.

The cache management algorithms were designed to provide the best overall performance in typical data processing installations and to protect all application data from loss due to power failure in the volatile storage. The first step was accumulating detailed information about patterns of access to data. Trace information was collected from a number of representative large MVS systems. Data reduction programs and analytic models were built for various alternatives. Through an iterative process, a set of

algorithms was developed and implemented in the microcode of the 3880 DASD control unit.

The procedure outlined here illustrates the thought process. The fundamental idea of the cache is to write a DASD track to the cache when the first read command for a record on the track is issued, in order to take advantage of locality of reference. The most obvious choice is to stage the entire track to the cache at this time. This raises the question of effectiveness from a performance standpoint. One consideration is that the CPU not be kept waiting while an entire track is read or staged into the cache. Thus the DASD is positioned to the desired record, and the record is transferred to the processor. At this point, one or more of the following actions are taken:

- Simultaneously read this record to the cache and to the processor.
- Continue reading the remainder of the track to the cache.
- End the read operation at the end of the track.
- Continue reading past the *index point* (the beginning of the track) and read the front part of the track to the cache (that is, the part of the track before the record originally requested).
- Reposition and read from the index point to the end of the track into the cache.
- Reposition to the record initially requested and read from it to the end of track into the cache.

The choice of which of these actions to select depends on modeling results and hardware/microcode capabilities.

Modeling results showed that reading the entire track into the cache was not the most effective action because there would be a penalty for control-unit-busy time due to staging the beginning portions of the track. Subsequent I/O requests would not compensate for this performance penalty. The action chosen was that of staging only data from the record requested to the end of the track. There were, however, further choices to be made. The hardware of the first IBM *cache control unit*, the 3880 Model 13, did not allow simultaneous reading from DASD to channel and DASD to cache. Therefore, after the record was transferred to the processor, the DASD had to be repositioned at the record requested and the rest of the track read into the cache. The follow-on cache control unit, the 3880 Model 23, was designed for simultaneous transfer between processor and cache. Thus it has achieved significantly better performance than its predecessor.

This same process of identifying the different alternatives for the operation of the cache, modeling the resultant performance, and fitting it to the hardware and microcode capabilities was repeated until the full set of functions was developed.

Another consideration is that the integrity of data must be preserved. This means that whenever a record is updated in the cache, the updated record must be written out to the DASD before the control unit signals the processor that the record has been written. The signal is called *device end*. Data integrity is ensured by having the control unit simultaneously write the update to cache and to DASD, as discussed later in the section on cache management.

Whether a record was fetched from the disk or from the cache, it is presented to the processor as though it came from the disk.

Regardless of whether a record was fetched from the disk or from the cache, it is presented to the processor as though it came from the disk. This eliminates the need for application code changes and minimizes changes to operating system code. To go a step further, data on a disk behind a cached control unit are identical in format to data on a noncached subsystem. This means that, in many cases, a conversion to cached control units is complete when the noncached control unit is replaced by a cached control unit.

Changes to the operating system are required to support the cache and have been incorporated into new releases of the operating systems (MVS/370, MVS/XA, and VM/HPO). These changes are in the DASD access methods, such as sequential and direct, and are described more fully in the section on cache management. Other changes include certain utility functions used primarily by system programmers and operations staffs, error recovery routines, and initialization code that builds the control blocks

describing I/O devices. Thus, these changes are entirely transparent to application programs using standard access methods. Furthermore, no change is required to the I/O GEN, which describes all of the devices attached to the system. Initialization code detects the presence of a cache and makes the appropriate changes to the I/O control blocks.

The design process had as a key requirement maintaining transparency of the cache.

The design process for the 3880 Model 13 and Model 23 had as a key requirement maintaining transparency of the cache to the system and the application programs. A special version of the Model 23, called the Record Caching RPQ, illustrates a different approach.[12] In this design, the performance advantages of a new function are significant enough to warrant changes to the application program. This version was specifically tailored for the Transaction Processing Facility, which is an operating system designed for very-high-performance transaction processing systems, such as airline reservation systems.

Because of the way the Transaction Processing Facility systematically spreads related records across multiple volumes, there is little benefit in staging a track of data when a particular record is requested. Instead, the cache storage is managed in slot sizes predetermined by the system programmer at the initialization of the cache by the operating system. When a record is requested, it is allocated to one of these slots and read into cache storage. Then normal least recently used (LRU) algorithms control its residence in cache. If data requested in an I/O operation are larger than any of the predetermined sizes, they are allocated whatever space is required in a special area of the cache storage. In this case, cache storage is used as a buffer only. After the record is used, the cache space is released. This implementation is called *record-level cache management.*

While not incorporated into access methods used by the Transaction Processing Facility, the RPQ does provide another new function, called *nonretentive data,* that can be incorporated into channel programs written for Transaction Processing Facility applications. In processing a transaction, these applications may make frequent use of transient records, or data to be used for the duration of a single transaction only. If there should be a system failure during the transaction, the transaction can be restarted, and the data can be recreated. Thus there is no need for forcing retention of the data on DASD. Nonretentive data can be used to handle the intermediate records. The I/O operation that writes such information includes a new channel command attribute that signals the record cache to omit DASD write-through and to keep the data in cache storage only. These data are not written to disk unless it is done by the LRU algorithm. This means that the initial write and all subsequent read operations are serviced out of the cache unless LRU algorithms destage the data. There is a significant improvement in performance; that is, these I/O operations could be accomplished in about 2.8 milliseconds, compared to the 20–25 milliseconds required for a 4K block of data to be written to disk.

Implementation

The 3880 Model 23 Storage Control is the first large-system component to use the new IBM 256K-bit dynamic random access memory chip. Use of this new technology allows for greatly extended cache sizes at very reasonable cost. There have also been significant enhancements to the 3880 Model 23 hardware and microcode, resulting in improvements in performance, reliability, availability, and serviceability. For example, for an eight-megabyte cache size, modeling results show that the 3880 Model 23 provides a 25 percent improvement in performance over the predecessor 3880 Model 13. At larger cache sizes, the improvement is even greater.[13]

The two cache control units are very similar in design and operation. This discussion concentrates on the architecture and operation of the 3880 Model 23. The main differences between the Model 13 and the Model 23 are identified in Table 1.

The 3880 Model 23 consists of two storage directors and 8, 16, 32, 48, or 64 megabytes of cache. Over 99 percent of the cache storage is used for storing copies of 3380 tracks, approximately 47K bytes in size. Known as *slots,* these are the unit of cache

Table 1 Comparison of 3880 Models 13 and 23

Feature	3880 Model 13	3880 Model 23
Technology	16K chip	256K chip
Cache sizes	4 and 8 MB	8, 16, 32, 48, and 64 MB
Transfer path	Serial	Branching
Dual frame	Yes	Yes
2-channel switch	Feature	Standard
Error detection and correction	Single-bit correct; double-bit detect	Detect all triple-bit errors and correct all double-bit errors and most triple-bit errors
Enhancements to improve availability		Single storage director caching Automatic reinitialization of cache
DASD supported	3380 Standard Models	3380 Standard and Extended Models
Software support	MVS/370 and MVS/XA; DFDS[1] and DFP VM/HPO Program Offering (Releases 3.2 and 3.4)	MVS/370 and MVS/XA DFP[2] VM/HPO[3] Program Offering (Releases 3.2 and 3.4)

[1] IBM Program Product Data Facility Device Support
[2] IBM Program Product Data Facility Product
[3] IBM Program Product Virtual Machine/High Performance Option

storage management. A very small part of the cache is used as a directory of control information about the data residing in the cache. The directory is used to translate the DASD cylinder and track information into the actual slot location in the cache. The amount of time required for this translation is negligible compared to the total I/O time.

The two storage directors may be in one 3880 Model 23, or they may be in each of two 3880 Model 23s in a dual-frame configuration, as depicted in Figure 2. The dual-frame installation option provides cross-control-unit configuration capability to enhance data availability.

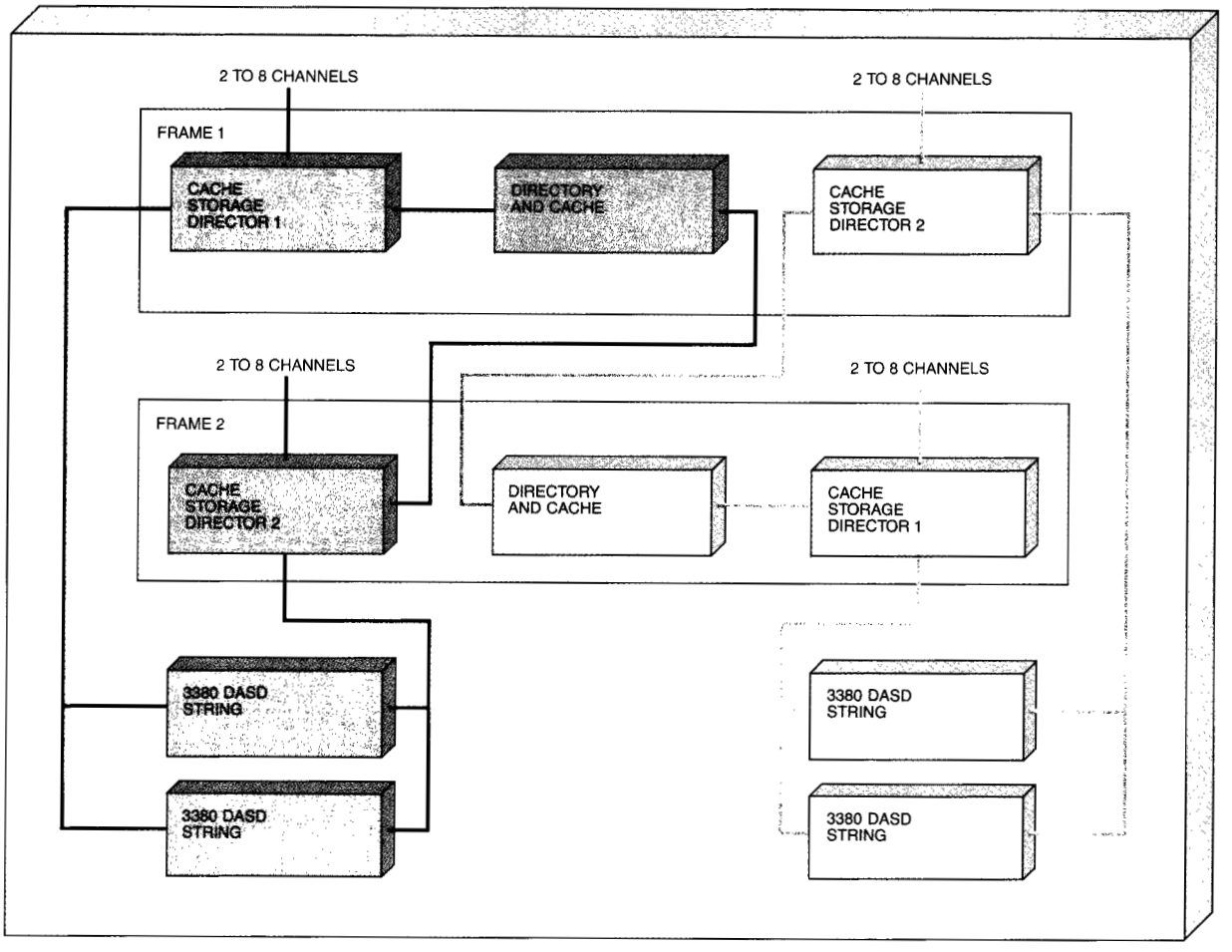
Cache management

A variety of methods are used to access data on disk devices,[4] and a number of these have been included as access methods that are provided with the operating system. These standardized methods include sequential and random forms of access. In a sequential process, the next record to be processed is also next in physical sequence on the DASD. In a random process, the next record to be processed need not be next in physical sequence. The three most significant random modes are the Basic Direct Access Method (BDAM), the Virtual Storage Access Method (VSAM), and the Partitioned Access Method (PAM). In BDAM, the application program contains code to determine where the next record will be placed on the disk device. Many sophisticated techniques (including randomizing routines) have been developed to efficiently fill and access data in BDAM files. VSAM imposes on the data an index structure that allows for fast and efficient access to the user's data. PAM is used for accessing information stored in a partitioned data set (PDS), the data set structure most commonly used for system and control information in an operating system. A PDS consists of a directory and a set of members, each containing data. To access information in a member, its name is specified, the directory is searched for the name, and the directory pointer is used to locate the data.

In each of these cases, the access method constructs a set of commands, called channel command words (CCWs), which are interpreted by the channel and result in channel commands that are sent to the control unit to carry out an I/O operation. These access methods have been modified to incorporate appropriate channel command words for the cache control unit. The CCWs vary according to the type of operation and what is most effective for that operation. This process is totally transparent to the applications programmer.

Several caching modes, including normal, sequential, and bypass-cache, are used to manage the cache storage.[14] These modes are set for the duration of an I/O operation, called a *channel program* or *CCW string*. The cache operates in normal caching mode unless directed otherwise by software. This is the most appropriate mode of operation for most access methods. *Normal caching mode* means that data are staged to the cache after being referenced in a read operation and remain in the cache until least recently used (LRU) algorithms allow the track slot to be overlaid with another track.

Figure 2 Two 3880 Model 23s in a dual-frame configuration



Sequential access methods [Basic Sequential Access Method (BSAM) and Queued Sequential Access Method (QSAM)] automatically instruct the control unit to process data using a *stage-ahead technique.* The first block requested is transferred to the processor and to the cache, followed by the remaining records to the end of the track. While the host is processing this track, the 3880 Model 23 automatically stages the next sequential track. As soon as the first data block of the second track is read by the processor, the first track slot is invalidated and the third track is automatically read into the slot previously occupied by the first track. This process continues through all subsequent tracks in the data set, as shown in Figure 3. In this fashion, data are prestaged into the cache for fast processor retrieval.

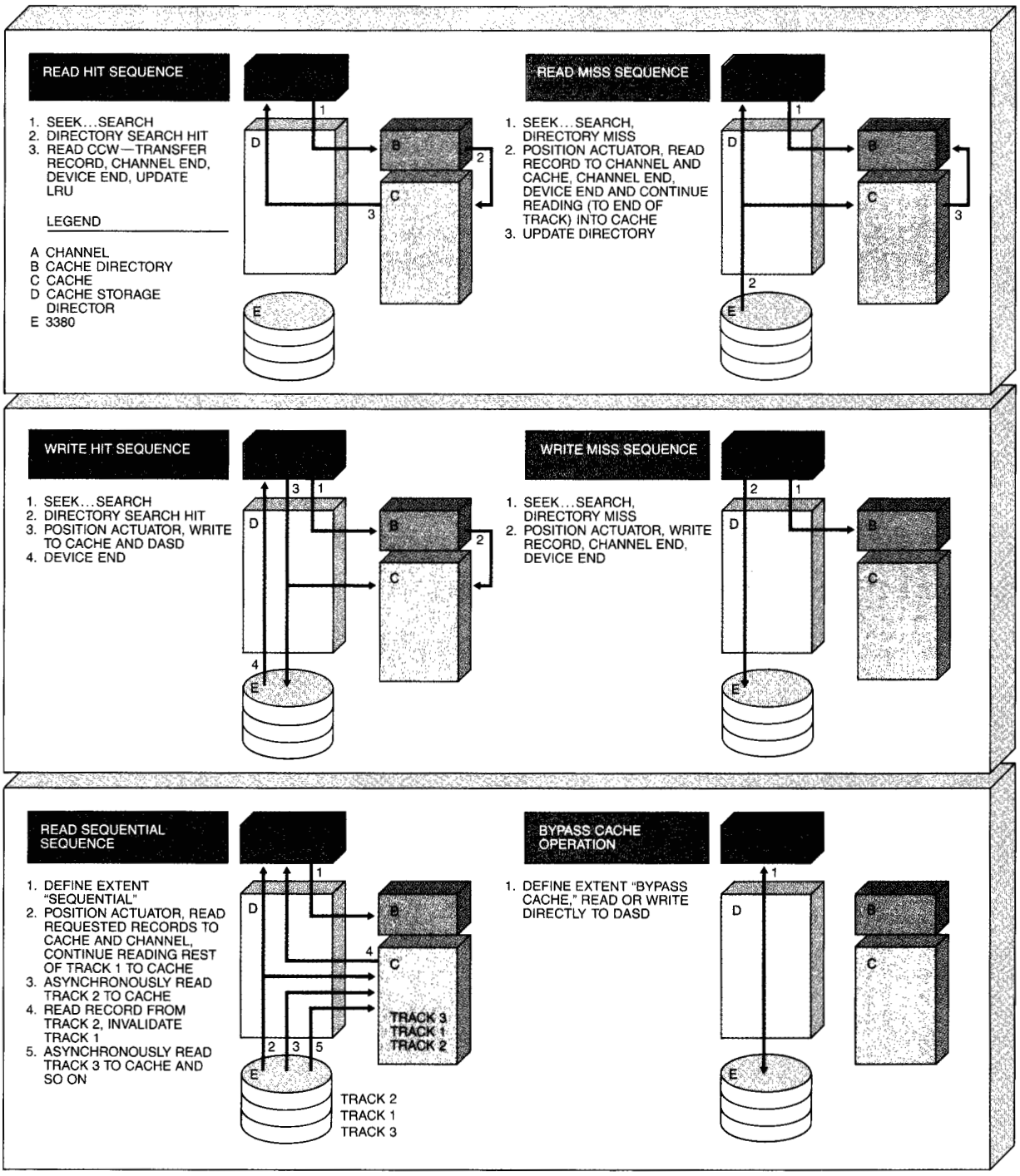
Another mode of operation is *bypass-cache,* where the cache is not used and the I/O operation goes directly to the 3380. This option is used, for example, by the dump/restore functions of the IBM Program Product Data Facility Data Set Services (DFDSS), where tracks are read once and not referenced again.

The 3880 Model 23 uses a branching data transfer to allow for simultaneous transfer of data from the channel to DASD and the cache, and from the DASD to the channel and cache. This capability provides a substantial reduction in control unit and device busy time for certain operations and a significant improvement in performance. Figure 3 illustrates the basic read/write operations.

When normal caching is specified for an I/O operation, one of four conditions occurs:

- *Read hit.* A read hit occurs when data requested by the read operation are in the cache. The data

Figure 3 I/O processing in the 3880 Model 23



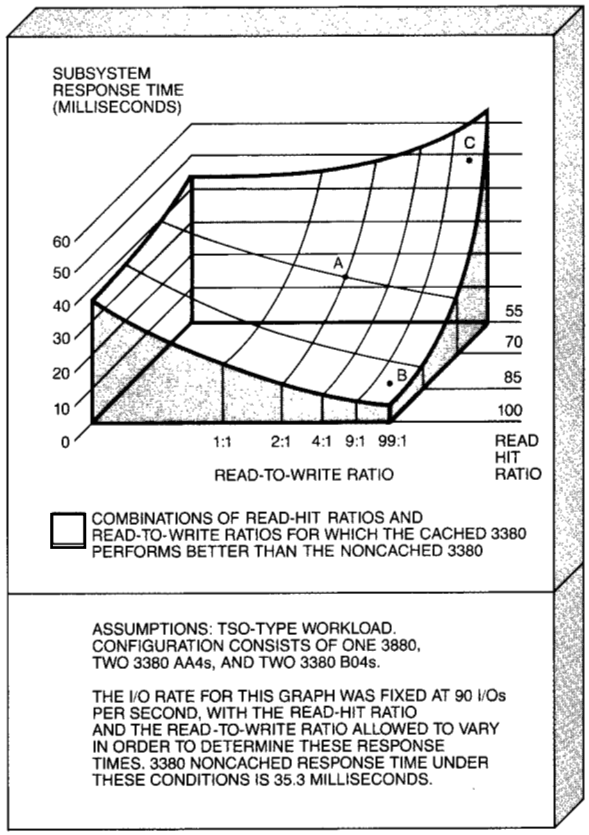
are immediately transferred to the processor at channel speed.

- *Read miss.* A read miss occurs when data requested by the read operation are not in the cache. In the Model 23, the DASD actuator is positioned and the record is read to the processor; at the same time, the Model 23 makes use of the branching data path to stage that record and all subsequent records to the end of the track into the cache. The Model 23 uses branching to permit overlapped data transfer operations and improves device and control unit utilization. Any new requests for these data result in read hits, assuming that the least recently used algorithm has not invalidated the cache copy of the track.
- *Write hit.* A write hit occurs when a record to be updated is already in the cache and on the 3380. The record in cache is updated because it may be referenced again. However, before the control unit can signal the operation as complete, it must ensure that the record has been successfully written to the 3380. This is another example of the use of branching by the Model 23. That is, the record is simultaneously written to the cache and to the DASD, with the *device-end* signal returned at completion. Write-through to the 3380 provides data integrity because copies in the cache and on the 3380 are identical.
- *Write miss.* A write miss occurs when the record to be written is not in the cache. The record is written directly to the 3380.

Read-hit and read-to-write ratios

The purpose of a cache control unit is to improve DASD performance. The fastest operation a cache can perform is a read hit. Therefore, the data selected for inclusion in a cache subsystem should maximize the percentage of read hits within reasonable limits. Two indicators that are very good predictors of cache performance are the *read-hit ratio* and *read-to-write ratio*. The read-hit ratio is the ratio of the number of read operations for which the record is found in the cache to all read operations. The read-to-write ratio is the ratio of the read operations to the write operations. The relationship of the read-hit ratio, the read-to-write ratio, and subsystem response time is shown in Figure 4. This graph shows the effect on subsystem response time of changing both the read-hit ratio and the read-to-write ratio, given a fixed I/O rate of 90 I/O operations per second through the cache control unit. The greater significance of the read-hit ratio can be seen where the response-time surface dips at the higher read-hit ratios, as shown at point B in Figure 4. Increases in the read-hit ratio have a much greater influence on response time than corresponding changes in the read-to-write ratio. Figure 4 compares the effects of cached and noncached implementations.

Figure 4 Relationship of read-hit ratio, read-to-write ratio, and response time of a cached 3380 subsystem



For good performance, the recommended read-hit ratio is 70 percent or greater. The read-to-write ratio should be greater than or equal to 2:1 in most cases, as shown at point A in Figure 4. Point B illustrates a very good combination of high read-hit ratio and good read-to-write ratio. If the read-hit and read-to-write ratios for the whole subsystem fall significantly outside the guidelines, performance may potentially degrade, depending on the system workload characteristics. An exceptional read-to-write ratio does not compensate for poor read-hit ratios, as shown by point C in Figure 4.

Cache installation

In large data processing installations, any change of hardware or software must be carefully planned and

scheduled so that the effect of the change on the users is minimized. Just as for any other change to the system, installing a 3880 cache control unit requires the following steps:

1. Planning for and installing software
2. Planning for and installing hardware
3. Establishing operational procedures
4. Establishing servicing procedures
5. Selecting data to be cached
6. Migrating data
7. Establishing techniques for measuring the results

Selecting data. The only task in this list with significant new activities is that of selecting data to be cached.

The 3880 Model 23 with its improved internal performance and particularly its larger cache sizes can provide improvements in DASD subsystem performance in many environments, without significant effort in selecting data. Table 2 shows specific types of data sets that are known to be good performers with a cache control unit. When a 16-megabyte or larger 3880 Model 23 is installed, data sets in the category of best cache candidates require little analysis beyond ensuring that the I/O rate through the control unit is reasonable. An approach is to start with a selection of data sets with a total of 70–100 I/O operations per second during the installation's peak period. More data sets and activity are added gradually, making sure that DASD subsystem response times and system-level performance remain acceptable.

The most quickly and easily realized system benefit is obtained from caching data catalogs, PDS, control data sets, and indexes.

Understanding the usage characteristics of good cache candidates can allow even more effective use of the cache. In addition, these guidelines can also be of benefit when evaluating either new applications or applications other than those listed in Table 2.

In the MVS environment, the following are two approaches to the cache data-selection process: one is intuitive, building on installation experiences; the other is analytic, using measurement tools such as the Cache Analysis Aid (CAA), which is an IBM Aid.

To provide optimal system performance improvement, data sets should be selected on the basis of three characteristics. The primary characteristic is the importance of the data to total system performance. Another characteristic is the locality of reference of the data for read operations, which is measured by the read-hit ratio. The third characteristic is the read-to-write ratio of the data.

Installation experiences show that the most quickly and easily realized system benefit is obtained from caching data sets such as catalogs, PDS, control data sets, and indexes. One common characteristic of these data sets is that they are used directly or indirectly by most users or functions executing in the system. Two examples are the RACF control data set, which must be referenced before any user is allowed to access data, and data in the Access Control Block Library (ACBLIB) of IMS, which must be referenced by each on-line transaction. RACF, the Resource Access Control Facility, is an IBM Program Product that facilitates data set security. The Information Management System (IMS) is an IBM data base/data communications system.

The best-performing data sets share the following usage characteristics: control-type data, high level of activity, good read-hit ratio, and good read-to-write ratio. Typically, the best performers are control or reference data sets rather than user data sets, and contain information necessary for the proper function of the system. The level of activity of these data sets is high, and there are usually many read operations for each write operation, because these data sets tend to have relatively infrequent update activity. Locality of reference is good, and certain groups of records tend to be reused frequently. Table 2 highlights good cache data sets, as experienced at a number of installations. Other candidates can be found by examining the contents of volumes with high I/O rates, high device utilization, and long queues.

Caching sequential data sets that account for substantial portions of the I/O activity for a job may result in reduced overall run times, assuming the quantity of data transferred per I/O to be less than half of a 3380 track.

Table 2 Data selection guidelines

	MVS/370 and MVS/XA			
	Systems/General	**TSO**	**IMS**	**CICS**[1]
Best Cache Candidates	Partitioned data sets PROCLIB Control data sets: RACF; HSM Catalogs, both ICF and CVOL	Libraries ISPF TSO user data Catalogs Volumes with heavy VTOC use	ACB MFS PROGLIB ADF work data base ADF rules[2]	PROGLIB
Good Candidates	Sequential input data sets (if of smaller block size)			
Potentially Good Candidates[3]	Other system-type data sets Shared data sets Look-up tables or dictionaries Custom systems		Image copy Data base indexes Smaller, primarily inquiry data bases IMS short message queue RECON data set Larger data bases (using Model 23 large caches) Scratch pad area Long message queue QBLKS	CICS application data sets
Not Good Cache Candidates	Write-only data sets Sequential data sets with half-track or larger block size Page and swap data sets			
	VM/HPO CMS[4]			
Best Cache Candidates	S and Y disks Installation-provided minidisks containing heavily used programs Read-only minidisks with multiple copies			

[1] Customer Information Control System
[2] Application Development Facility
[3] Custom-designed systems or other software may have good cache candidates that should be evaluated by the criteria given in the section on selecting data.
[4] Virtual Machine/High Performance Option Conversational Monitor System

Certain candidate volumes may be validated by using the Cache Analysis Aid, which analyzes GTF CCW trace data to report read-hit ratios, read-to-write ratios, I/O rates, and data transfer sizes of the potential caching volumes. The GTF CCW trace is an MVS facility for logging all channel programs to a specified set of devices. Volumes should be added or deleted in successive runs until the overall hit ratio is more than 70 percent, the read-to-write ratio is 2:1 or higher for the proposed cached volumes, and the total I/O rate for the control unit is in the range of 70–100 I/O operations per second. Typically, this will take only two or three iterations of the CAA. The data sets and volumes to be moved under the cache should be prioritized, placing the most important ones there first. After initial installation and performance assessment, the volumes being cached may be adjusted to increase or decrease this I/O rate, depending on system performance criteria. It is more efficient to start conservatively and then add more activity to the cache, rather than starting with high activity and then reducing it.

Because the 3880 Model 23 processes write operations efficiently and because of the improved internal operations, even though individual volumes may fall outside these guidelines, they may still show performance improvements with the cache. Two such

examples are the IMS short message queue and the Application Development Facility (ADF) work data bases. Volumes with hit ratios as low as 60 percent and read-to-write ratios of about 1.5:1 have shown improvement. These should be provisionally placed behind the Model 23 and evaluated for acceptable performance.

Additional tuning techniques. The presence of cache control units gives us a new option for DASD performance tuning, i.e., increasing the size of the cache storage. When larger caches were installed, several early installations of the 3880 Model 23 demonstrated the ability to cache entire strings of DASD. Increasing cache storage should be considered as an alternative to extensive, ongoing monitoring and tuning of a cached subsystem.

The basic principle of DASD tuning is to balance the critical I/O load across paths and separate high-activity or high-contention volumes on different paths whenever possible. This principle also applies to cached DASD, with some of the following variations made possible by its high-performance characteristics:

- *Balance activity.* When multiple 3880 cache control units are installed, overall I/O rates across all cached control units should be balanced among the 3880 cache control units. Next, cached I/O rates should be balanced across these units. Data set types should be mixed. For example, all PDSs should not be placed together, nor should all catalogs. Providing a mix of data sets or functions usually results in better performance. Cache control units substantially reduce the effect of shared DASD, so that separating shared data sets is not as significant an activity with the cache. However, very heavily used data sets with significant reserve/release activity should be placed on different volumes.
- *Use cache selectively.* Optimal system and 3880 cache control unit performance can be attained by caching a relatively small percentage of the I/O operations. Caching only the most important data increases the likelihood that the records requested will be in the cache and reduces the contention for the cache resource.
- *Schedule changes.* Optimization of the cache configuration for specific workloads can be accomplished by changing the cache configuration for major workload changes. Transition between cache configurations is accomplished by using the Access Method Services command SETCACHE, which is described later. For example, caching data base index volumes during overnight batch processing may improve batch elapsed run time, while caching the IMS data sets ACBLIB and MFSLIB online during the day may help improve terminal user response time.
- *Consolidate highly active, good caching candidates.* A number of high-activity and/or high-contention data sets may be consolidated on one storage director pair, because of the higher I/O rate and faster response of the 3880 cache control units. The extent to which this can be done is determined by the characteristics of the data sets and the level of activity within the storage director pair.

MVS considerations. The primary tools for monitoring cache performance in an MVS system are the IBM Program Product Resource Management Facility (RMF) and the Cache RMF Reporter program offering.[15] On a day-to-day basis, the key RMF report is the Device Activity Report, which reports I/O rates, device service times, and other DASD performance statistics. As long as the device service times and the response times stay in an acceptable range, no other monitoring is usually required.

Control of which minidisks are cached is provided by the directory.

The hit ratios and the other values unique to a cached 3880/3380 environment then require study only on a performance-exception basis. One approach is to use the Cache RMF Reporter program offering, which uses an RMF exit to collect cache statistics and a postprocessor program to report cache activity, including hit ratios and read-to-write ratios for all cached volumes and the entire subsystem. Even though the RMF exit data might be used only occasionally, they should be archived with the rest of the RMF data because they can be valuable for capacity planning.

In addition to the changes to access methods for supporting cache functions, Access Method Services

(AMS), a collection of data set utility programs provided with VSAM, has new commands used for cache communication. SETCACHE is used to set volumes on or off to the cache, as well as to activate and deactivate caching for the entire subsystem. SETCACHE permits changing the caching configurations, depending on production workloads or data selection decisions. LISTDATA is used to gather activity statistics and subsystem status. A new option of LISTDATA provides the capability of displaying cached subsystem status at the operator console.

VM considerations. All of the data selection and configuration guidelines just given apply to the VM/HPO CMS Program Offering environment as well.[16] Caching is set on and off at the volume level with the CACHE command. Control of which minidisks are cached is provided by the directory. Minidisks are cached selectively, with minidisks with high use, good locality of reference, and a good read-to-write ratio being the best candidates. These include the S and Y disks. The VM Monitor may be used to identify candidates and collect performance data. It may be desirable to isolate high-activity minidisks to their own volumes. Cache configurations may be altered with the VM operator command CACHE.

Performance of the cached 3380

I/O performance. The capability of IBM processors has been increasing at a compound annual growth rate of over 40 percent. Improvements in DASD performance, although very substantial, cannot match the internal processor speed without the assistance of an electronic device, such as a cache. The 3880 Model 23 is a strategic element in a total hardware/software system, which further improves 3380 performance. This systems approach includes use of the following techniques:

- *Add more data paths and devices.* Enhancements in processor architecture and MVS/XA allow the effective use of more paths and devices, which can result in improved performance by reducing path and device contention. In some cases, however, it may be neither practical nor cost-effective to spread the data across multiple paths using non-cached DASD. Although an increase in the number of paths may reduce contention for data sets, subsystem response time is limited by mechanical motion, that is, the seek, latency, and RPS miss characteristics of the DASD.
- *Increase data path utilization.* The Dynamic Path Reconnect feature of MVS/XA enables higher channel utilization. Again, the limiting factor in this approach is the mechanical nature of the DASD.
- *Install faster devices.* Install the fastest available DASD subsystem.

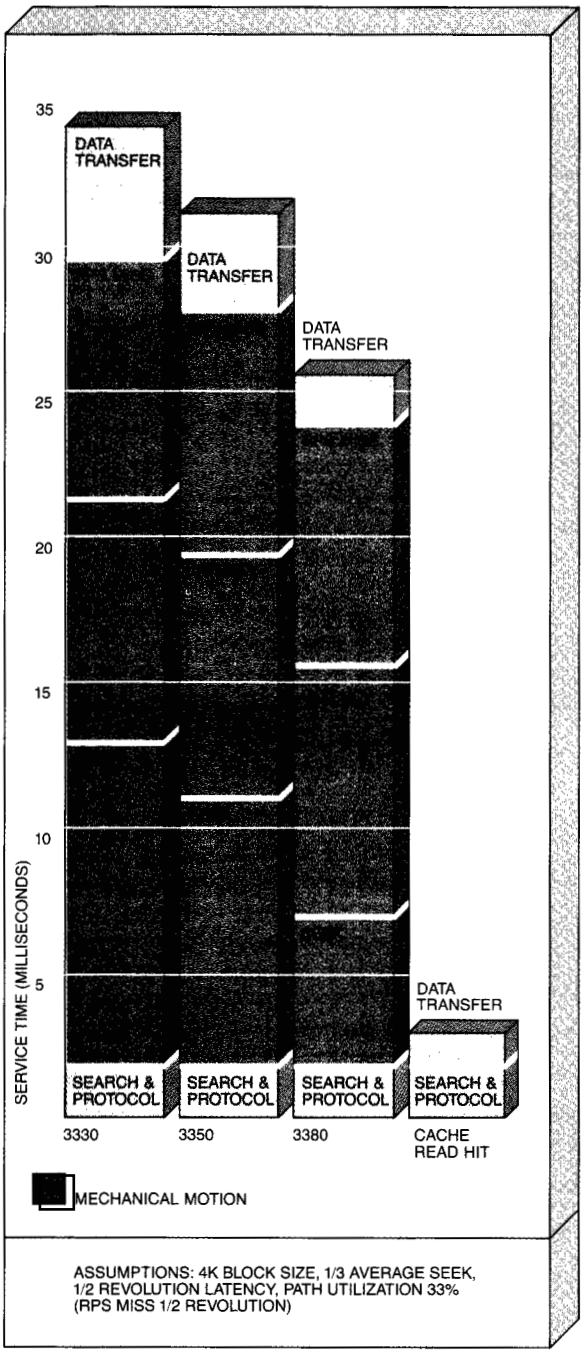
Improved service times. Improvements in IBM DASD service times are illustrated in Figure 5, which presents the major components and typical timings of a DASD read operation. One can see here that the greatest improvement is attained with the recent introduction of cached control units. DASD seek, RPS miss, and latency are eliminated when the data are read from the cache. Although there will be continued improvements in DASD electromechanical operations, it does not seem likely that electromechanical operations will reach the speed of an electronic cache.

Most installations are expected to experience some combination of increased I/O rate and better device response time.

Higher I/O rates. Figure 6 shows modeled performance curves for 3380 DASD and a 3880 Model 23 with a 16-megabyte cache. In this example, the analytical model uses a workload with read-hit ratio, read-to-write ratio, and access skew derived from characterizations of large customer TSO systems. The curves plot the 3380 subsystem response time as a function of the access rate. As can be seen from the graph, the subsystem response time increases as the access rate increases. This is due to the fact that at the higher access rates, requests are arriving faster than they can be handled, so that queueing delay becomes a component of the DASD subsystem response time.

Replacement of a 3880 Model 3 by a Model 23 causes a substantial improvement in the performance of the subsystem. If we assume constant I/O rate and mix of activity, the result will be better response time, as shown by point A on the graph. If we assume a significant increase in I/O activity, the same level of response may be experienced, but at a much higher I/O rate (point B). Most installations

Figure 5 Improvements in average read service times for IBM storage devices



are expected to experience some combination of increased I/O rate and better device response time.

The experience of a large IMS/DC installation with the 3880 Model 13 illustrates both possibilities. After installation of the 3880 Model 13, there was an immediate improvement in the IMS transaction rate of about 10 percent, accompanied by faster DASD response. Over the next six months, the IMS workload increased another 15 percent, with no additions to the I/O subsystem. The installation made the transition to the higher performance level of the 3880 Model 13, and then moved up that curve as additional demand developed.[17]

Because of the ability of the 3880 Model 23 to sustain higher I/O rates, greater numbers of active data sets can typically be placed on volumes controlled by the Model 23, while maintaining the same level of subsystem performance. *Access density*—the term that expresses this relationship—is the ratio of accesses per second to DASD capacity in gigabytes. Figure 7 shows that at a response time of 25 milliseconds, a 3880 Model 23 with 16 megabytes of storage, for example, can sustain more than twice as many I/O operations per second to a gigabyte of data than can a Model 3.

From a user's viewpoint, improved access density means that heavily used data sets are no longer a major performance or tuning concern. A popular technique for reducing the level of activity and the contention for a data set has been to make several copies and to direct subsets of users to particular copies. The improved access density of the 3880 cache control units reduces the need for multiple copies or split data sets. As an example, one cache user was able to consolidate multiple RACF data sets onto one volume shared by two systems, while sustaining 25 I/O operations per second with a response time of about 10 milliseconds.[17]

Cache as an alternative to short DASD strings

The performance tuning of a computer system can realistically never be done without considering the prices of the various alternative systems. Clearly, if price were no object, systems performance specialists would have an easy job. Cache control units now change the former way of thinking about costs of improved DASD performance. One way this happens is that the cache is capable of delivering many more I/O operations per second at a given level of response than can a noncached control unit. This is illustrated

by a comparison of the use of short DASD strings versus installing a cache.

The significance of access density and price/performance can best be demonstrated by a comparison of the performance of short DASD strings with the performance of a 3880 Model 23 cache subsystem. One of the most commonly used means of improving noncached DASD subsystem performance has been to increase the number of paths to the DASD and use fewer actuators per path, i.e., to use short DASD strings. Increasing the number of paths and using shorter strings reduces the amount of contention in the DASD subsystems and can thereby improve performance. However, modeling studies of the two different approaches show that the best way to improve performance is to add cache.

Figure 8 shows the configurations used in the model runs. All configurations have 32 actuators. The modeling results show the effect of changing paths or adding cache. The base configuration consists of two 3880 Model 3s and two full strings of 3380s. In the short-string configuration, two 3880 Model 3s were added, and the 3380 strings were made half as long to provide four more paths and shorter strings. The 3880 Model 23 configuration has two cached control units with 16 megabytes of cache storage and two strings of 3380s. All three model studies use the same TSO workload as was used in the previous studies. In particular, the read-hit ratio of 91 percent, the read-to-write ratio of 4:1, and the DASD skew were derived from large customer TSO systems.

Comparing the performance curves of Figure 9 illustrates the significant effect of the higher access den-

Figure 6 3880 Model 3 and 3880 Model 23 access rate versus subsystem response time

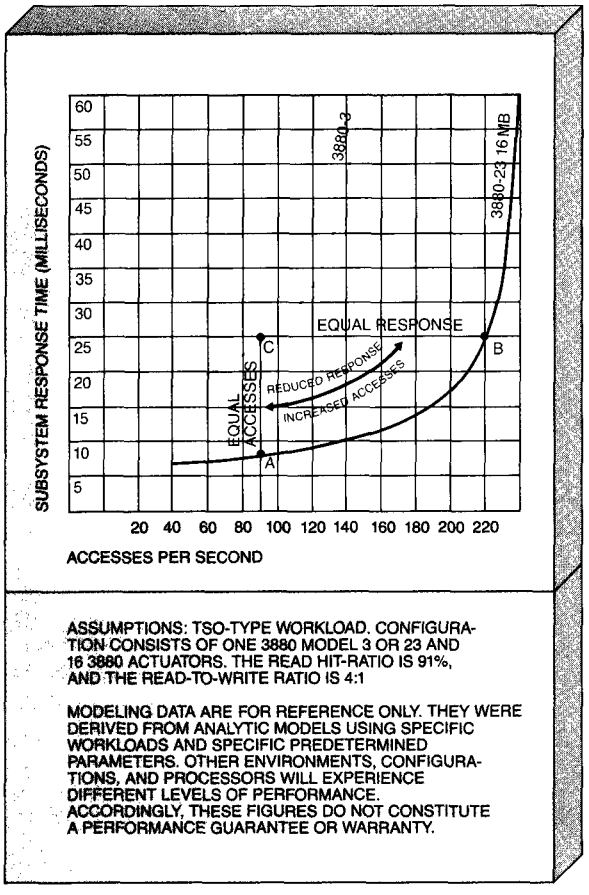


Figure 7 Access density of 3880 Model 3 and 3880 Model 23 subsystems

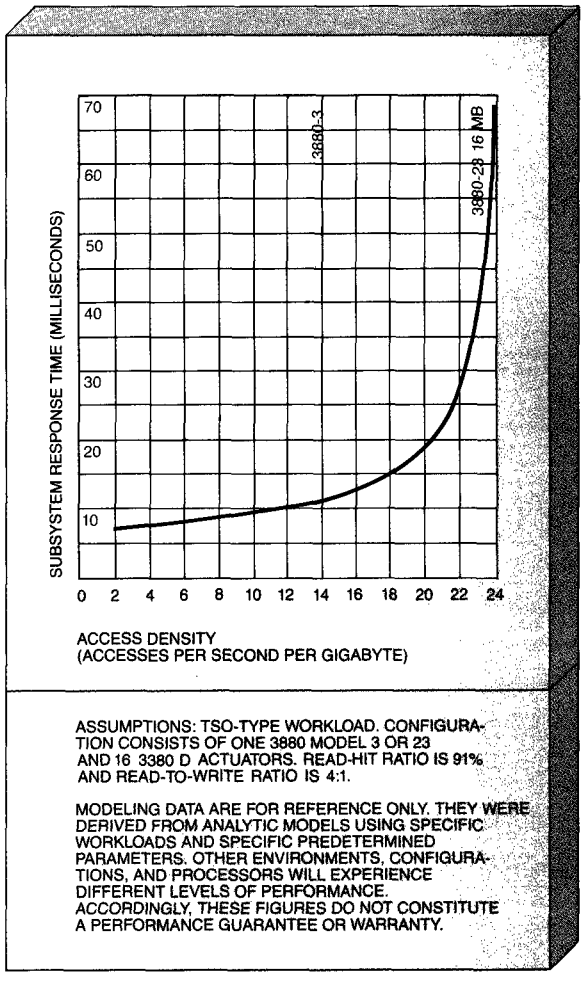
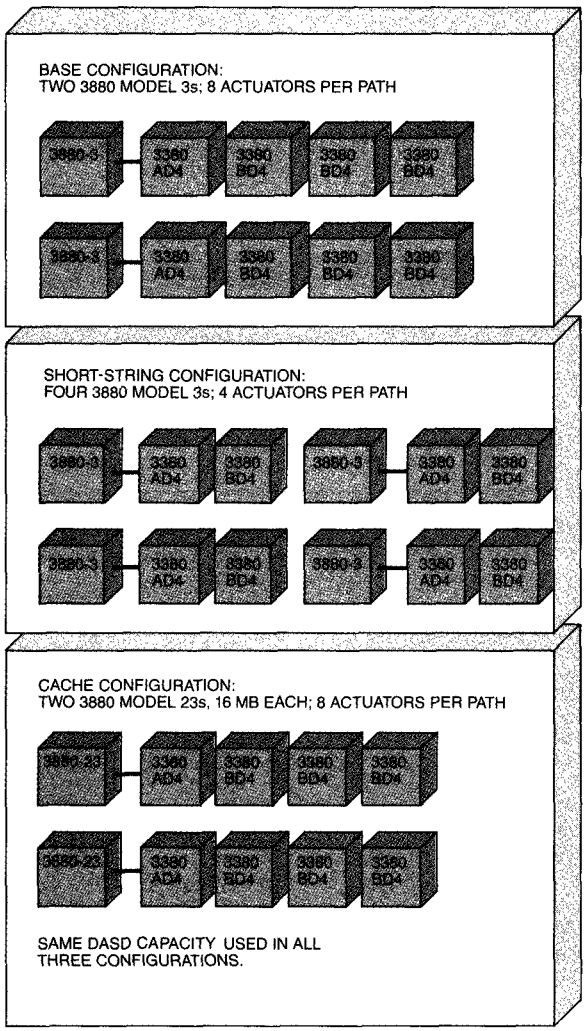


Figure 8 String length comparison: modeled configurations



sity and better price/performance of the 3880 Model 23. When the 3880 Model 23, with its substantially better access density, is introduced into the subsystem, there is a marked improvement in performance compared to the cost of the configuration. For example, at a 25-millisecond response time, the 3880 Model 23 with 16 megabytes of cache provides a performance improvement of over 200 percent for an additional cost of 35 percent for the subsystem. In contrast, the short-string configuration can provide only a one-third performance improvement for its additional cost of 26 percent. In this environment, the use of the 3880 Model 23 provides about five times the price/performance of the short-string configuration and provides for subsystem growth beyond the capability of other configurations.

It is important to note that this example shows a 3880 Model 23 configuration operating at an access rate of approximately 165 I/O operations per second per control unit. At lower I/O rates, a cache configuration provides better response time than a non-cached configuration. Several installations have experienced control unit average response times in the 11- to 20-millisecond range, with 50 to 85 I/O operations per second through the control unit. Such response times are below the capability of a non-cached configuration at comparable I/O rates.

Customer experiences

Summarized here are experiences of customers with the 3880 Model 13 and Model 23. The 3880 cache control units have been used in a number of ways by many installations to provide increased I/O rates and better service times. Performance and system improvements may be summarized as follows:

- *Improved end-user service.* Large TSO systems have benefited significantly from caching TSO user data sets, libraries, catalogs, and RACF control data sets. Typical improvements in user transaction rates have been about 6 percent, and response-time improvements have been in the range of 13–27 percent. Large IMS systems have seen improvements in user response time of up to 30 percent, transaction rate increases up to 25 percent, or some combination of improved response times and transaction rates. Other benefits experienced have been more consistent response time during periods of peak activity and the ability to absorb the latent demand represented by new users.[17]

- *Reduced I/O contention.* High-use and shared data sets are often a source of concern because they may require a significant amount of performance tuning. Substantial relief in this area can be realized. At one installation, response time to a system link library regularly exceeded 70 milliseconds. At the same installation, but with the 3880 Model 13, response time was reduced to about 15 milliseconds. In some installations, several copies of such critical data sets as IMS program libraries and link libraries are often maintained to reduce contention and I/O rates. With the cache control units, many of these copies can be deleted, an advantage from both performance-management and maintenance standpoints. A VM/SP HPO installation with two processors and four copies of its S and Y disks was able to consolidate these to one copy of each, shared between the processors through the

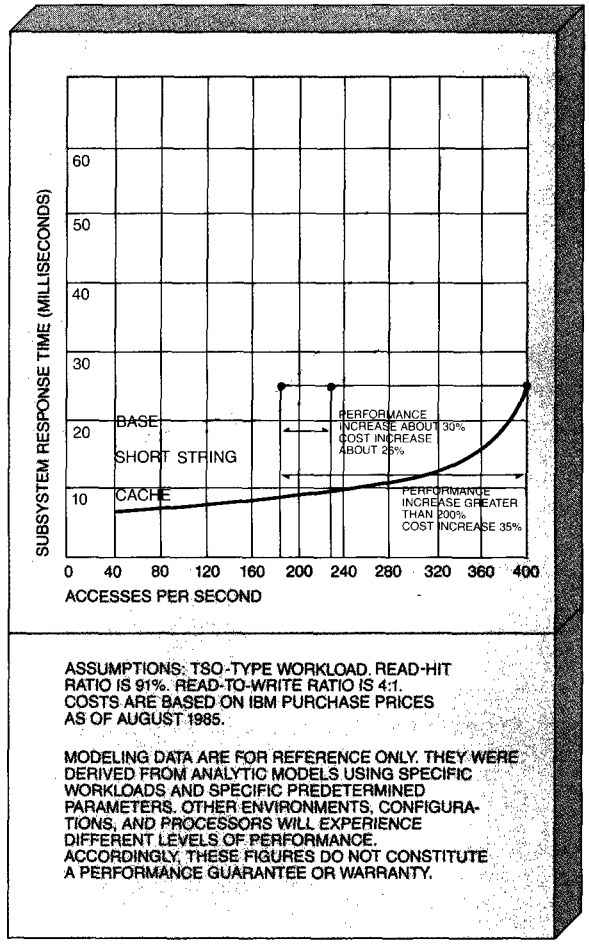
use of the cache. At the same time, DASD response time dropped by 60 percent and the I/O rate increased by 10 percent. Where RACF data sets were split and distributed across different paths, they have been consolidated behind the cache control unit and have exhibited improved response times at higher access rates. For example, a very large development system has a RACF control data set with a 5–7-millisecond response time and a sustained I/O rate between 32 and 40 I/O operations per second. Catalogs are also very good cache performers. Catalog volume service times in the 6–12-millisecond range with I/O rates in the range of 15–18 I/O operations per second have been attained.[17]

- *Increased I/O rates.* With appropriate data (good hit ratios, high read-to-write ratio, and the appropriate mix of types of accesses), the 3880 cache control units have sustained significantly higher I/O rates, both to an individual actuator and to the subsystem, than noncached 3380. For example, one batch DL/I bill of materials application sustained an I/O rate of 60 I/O operations per second to a single volume over a four-hour period. A large TSO system has routinely experienced 150 I/O operations per second through the 3880 Model 13.

  The 3880 Model 23 Storage Controls have run at rates of from 60 to more than 200 I/O operations per second. Compare this to the 3880 Model 3 control unit, which is typically in the range of 50–80 I/O operations per second, or higher with well-tuned systems. The cached I/O rate is usually substantially higher than the same configuration could sustain without the cache. All of these rates reflect measurements over periods of time extending from several hours up to a full on-line day. For periods of time measured in minutes, any of these rates can be much greater. The ability to sustain higher I/O rates has enabled some installations to lengthen their 3380 strings behind the cache control units without any degradation in response and with reductions in cost and floor space compared to alternative configurations.[17]

- *A base for system growth.* Many customers are experiencing a period of great application growth. The 3880 cache control units are system components that have eased this growth, allowing installations to respond effectively to their changing environment after installing a cache. One IMS/DC installation described in more detail earlier in this paper was able to sustain a 25 percent growth in user transactions without new hardware or reconfiguration.[17]

Figure 9 3880 Model 23 and short-string DASD comparison



These experiences illustrate the effectiveness of a cache implementation for disk devices in contributing to higher system throughput capability and improved user service. Such improvements may constitute faster end-user response times and higher transaction rates, higher system utilization, and quicker turnaround for batch jobs.

Concluding remarks

In summary, the cache control unit is the most recent device designed to improve computer system performance by providing data to the processor at a higher rate. While it contains a number of innova-

tions in its implementation, this control unit also incorporates ideas developed over the years to improve system performance. The cache control unit is composed of new hardware and microcode, supported by appropriate modifications of the operating systems. Taken together, the cache control unit and its innovations provide improved DASD subsystem performance. The ability of the cache control unit to allow higher system use, alleviate systems tuning problems, and provide better end-user response time has been well demonstrated in a large number of customer installations.

Acknowledgments

I would like to thank the customer personnel and IBM systems engineers who shared their cache experiences with me over the last three years. Their collected experiences formed the basis for developing the data selection guidelines and tuning techniques contained in this paper. I also thank Marc E. Goldfeder for sharing with me his extensive analytic modeling work on the cache control units. Finally, I thank S. J. Caldwell, J. H. Cord, G. L. Grossman, M. H. Hartung, D. A. Johnson, J. Kranz, P. Y. Pang, and R. Wicks for their technical assistance.

Cited references and notes

1. H. H. Goldstine, *The Computer from Pascal to von Neumann*, Princeton University Press, Princeton, NJ (1972).
2. L. D. Stevens, "The evolution of magnetic storage," *IBM Journal of Research and Development* **25**, No. 5, 663–676 (1981).
3. R. E. Matick, *Computer Storage Systems and Technology*, John Wiley & Sons, Inc., New York (1977).
4. M. Bohl, *Introduction to IBM Direct Access Storage Devices*, Science Research Associates, Chicago (1981).
5. J. M. Harker, D. W. Brede, R. E. Pattison, G. R. Santana, and L. G. Taft, "A quarter century of disk file innovation," *IBM Journal of Research and Development* **25**, No. 5, 677–690 (1981).
6. J. S. Liptay, "Structural aspects of the System/360 Model 85; II. The cache," *IBM Systems Journal* 7, No. 1, 15–21 (1968). The term *cache* is taken from the French word for a hidden storage place. Use of the term in the context of computer storage was suggested by Lyle R. Johnson, who was editor of the *IBM Systems Journal* at the time this paper was published.
7. *Introduction to the IBM 3880 Storage Control Model 23*, GA32-0082, IBM Corporation; available through IBM branch offices.
8. A. Padegs, "System/360 and beyond," *IBM Journal of Research and Development* **25**, No. 5, 377–390 (1981).
9. S. E. Madnick and J. J. Donovan, *Operating Systems*, McGraw-Hill Publishing Co., Inc., New York (1974).
10. E. G. Coffman, Jr. and P. J. Denning, *Operating Systems Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1973).
11. L. A. Belady, R. P. Parmelee, and C. A. Scalzi, "The IBM history of memory management technology," *IBM Journal of Research and Development* **25**, No. 5, 491–504 (1981).
12. *IBM 3880 Storage Control Record Cache RPQ #8B0035 Introduction*, GA32-0086, IBM Corporation; available through IBM branch offices.
13. Modeling data are for reference only. They were derived from analytic models using specific workloads and specific, predefined parameters. Other environments, configurations, and processors will exhibit different levels of performance. Accordingly, these data do not constitute a performance guarantee or warranty.
14. P. Jeremy, G. Lawrence, A. Medeiros, G. Neaga, and S. Wallace, *Guide to the IBM 3880 Storage Control Model 23*, GG24-1642, IBM Corporation; available through IBM branch offices.
15. *Cache RMF Reporter Program Description and Operations Manual*, SH20-6295, IBM Corporation; available through IBM branch offices.
16. *VM/SP HPO CMS Support for the 3880 Model 13 and 3880 Model 23 Program Description/Operations Manual*, SH20-6537, IBM Corporation; available through IBM branch offices.
17. These customer experiences were observed in specific installations. Results for other installations, workloads, and environments will vary and must be assessed on an individual basis. Accordingly, these figures do not constitute a performance guarantee or warranty.

General references

K. G. Dahman and G. L. Grossman, *Effective Use of Cached DASD in a Data Base/Data Communications Environment*, Technical Report TR-82.0095, IBM General Products Division, Tucson, AZ 85744 (1983).

R. A. Lindsay, *3880 Model 13 Performance Measurements*, GG22-9337, IBM Corporation; available through IBM branch offices.

K. G. Dahman, R. G. Edison, J. S. Hyde, W. J. Walsh, and L. T. Wang, *IBM 3880 Model 23 Performance Measurements*, GG66-0223, IBM Corporation; available through IBM branch offices.

Carol Porter Grossman *IBM General Products Division, Tucson, Arizona 85744.* Mrs. Grossman is an Advisory Planner for storage control systems planning. Since 1982, she has worked in the areas of marketing requirements and field support for the IBM 3880 Models 13 and 23. She joined IBM in Chicago in 1974 as a systems engineer. While there, she was a storage systems and Mass Storage System (MSS) specialist. She is the author of several articles on the installation and use of cache control units. She received her B.S. in 1967 and M.S. in 1968, both in mathematics, from Northwestern University, Evanston, Illinois. Mrs. Grossman is a member of Phi Beta Kappa and Pi Mu Epsilon.

Reprint Order No. G321-5257.