IBM

Washington Systems Center

Technical Bulletin

Paging Enhancements in VM/SP HPO 3.4

T. Beretvas
W. Tetzlaff

GG22-9367-00
May 1984

Washington Systems Center
Gaithersburg, Maryland
Technical Bulletin

Paging Enhancements in VM/SP HPO 3.4

Tom Beretvas
William Tetzlaff
Edited by William O'Brien

GG22-9367-00
May 1984

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis **without any warranty either expressed or implied**. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Performance data contained in this document was determined in a controlled environment; and therefore, the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to: Washington Systems Center, 18100 Frederick Pike, Gaithersburg, MD 20879.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

This document explains the changes in the VM/SP paging subsystem for VM/SP HPO Release 3.4. The motivation and the background for these paging enhancements are discussed. The design of the new paging subsystem is described and contrasted with the old design. Specifically, the following new functions are highlighted:

- logical and physical swapping
- block paging
- trimming
- pre-paging
- N-select
- moving cursor
- disposable page collection

The new SET commands and their effects are discussed.

The last section of the document discusses some of the things learned in running a prototype of the paging enhancements in a CMS interactive environment. It also answers some configuration questions.

Note: The material in this Technical Bulletin was originally prepared for an oral presentation at SHARE 60.5, Salt Lake City.

THIS PAGE INTENTIONALLY LEFT BLANK

This Technical Bulletin is being made available to IBM and customer personnel. It has not been subject to any formal review. The information herein has not been submitted to any formal IBM test and is presented on an **"as is" basis without any warranty either express or implied**.

The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Users of this document should verify the applicable data for their specific environment.

THIS PAGE INTENTIONALLY LEFT BLANK

Many hours were spent in the design of the paging prototype for VM/SP HPO 3.4.

The principal prototype development contributors are:

- T. Beretvas
- W. Buco
- G. Greenberg
- D. Patterson
- G. Spivak
- W. Tetzlaff
- L. Wheeler

The principal prototype measurement contributors are:

- R. Bos
- G. Lothringer
- T. Meggenhofen II
- J. Mitchell
- J. O'Connell

We know that even more hours were being spent by the developers at the DSD laboratories to take the paging subsystem prototype and combine it with other performance enhancements to build VM/SP HPO 3.4.

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ILLUSTRATIONS

THIS PAGE INTENTIONALLY LEFT BLANK

INTRODUCTION

VM/SP users improve productivity through lowered system response times (1 and 2). Analysis of VM/SP systems frequently shows that the greatest leverage in improving response times comes from improving the paging subsystem (3). The purpose of this paper is to explain the enhanced paging and real storage management algorithms of VM/SP HPO 3.4.

VM/SP HPO 3.4 contains many <u>other improvements</u> besides the paging enhancements. This paper only concerns itself with the paging enhancements.

We have extensively studied VM/SP paging (4) on a number of real systems. These studies indicate some problem areas, but more importantly they suggest that interactive users have repeatable working sets across transactions. These studies led to some experiments on real systems.

THIS PAGE INTENTIONALLY LEFT BLANK

Several factors prompted the creation of the VM/SP HPO 3.4 prototype.

The increasing emphasis on interactive response time led to an analysis of the response time itself. Many times the largest component of the response time is paging delay. This fact led to considering how paging response time could be improved. Our analysis of paging in CMS-intensive systems suggested that interactive users had repeatable working sets across transactions. We concluded that page reference and scheduler event information could be used to improve the page replacement algorithm. Further, the same information could be used for pre-paging.

We then looked at the tremendous growth in CPU power, "MIPs", which was not matched by a similar reduction in DASD paging access time. Note that 3380s reduce seek and data transfer times, but not latency. The apparent access time could be improved if multiple pages could be moved with a single access.

To reduce paging delays it is necessary to move pages into and out of main storage faster than the existing CP demand paging, or single page access environment, is capable of doing. One way of doing that is using transfer of multiple contiguous pages with a single SIO. The CPU cycles required for the paging SIOs and interrupts would be reduced.

During our analysis of the system we found several cases where users are unnecessarily dropped from queue during the life of a Q1 transaction. In the product the number of "false" Q-drops are reduced, the Q1 drop counts more closely correspond to transaction counts. Experience has shown that while the real response time is not affected significantly, the Q1 drop rate may be reduced by as much as two thirds of its previous value, and a corresponding three-fold increase in the VMAP Q1SEC value may occur.

THIS PAGE INTENTIONALLY LEFT BLANK

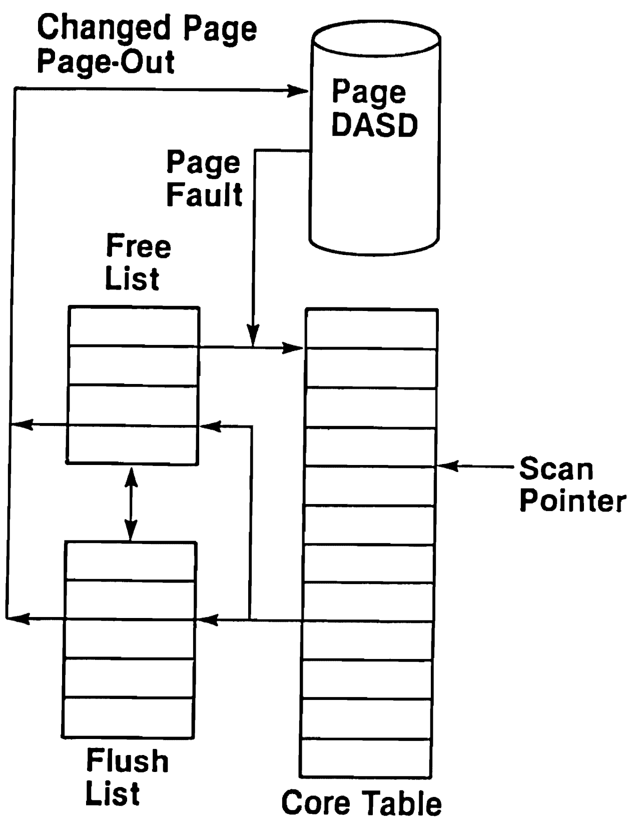

Figure 1. Basic VM paging flow

Figure 1 shows a conceptual image of paging in VM/SP and VM/SP HPO (prior to HPO 3.4). This is referred to as "VM" in the rest of this paper.

When a page frame[1] shortage is detected in VM, i.e. the number of free page frames plus number of pages being written out is less than a threshold, the "core table" scan mechanism is invoked to replenish the available frame list[2]. This cyclic scan of the core table looks for available page frames. Since the ordering of the core table is by real storage address, the scan affects users in a random fashion. The core table scan takes unreferenced pages in its path until the threshold is met. Since the usual shortfall is only one page frame, it is usual to acquire only one page each time. If the page taken was changed (from the previous copy on secondary storage), then it has to be written out onto a paging DASD, before the frame is made available for re-use. Thus, normally one page is written per SIO. Page frames containing unchanged pages are moved directly onto the free list. VM is a global LRU (Least Recently Used) demand paging system prior to VM/SP HPO 3.4. This means that most of the time one page is moved per SIO, either as a result of a page-in (page-fault) or page-out operation. Practically the only chance for multiple page transfers is if the paging device was overloaded, and therefore a queue of paging requests was built up. Under these circumstances CP combines the paging requests into a single SIO if the queued requests go to the same paging cylinder.

The free list in VM is a list of immediately available page frames. This list is kept at a size that is equal to the multiprogramming[3] set plus one. This insures that if each task requires one free page each in rapid succession then one free page will still be available. The free list is replenished whenever it falls below the threshold. Requests for frames come one at a time, because of page faults; so the free list tends to fall one below the threshold. Core scan then has to find one page for the free list; it takes one frame.

The first choice for refilling the free list is the flush list. In general, the flush list is empty, so that the second choice, the core table, is used. In well tuned systems flushing normally does not occur. Thus the flush list is not used.

The algorithm that is used to select a page for paging out is called the page replacement algorithm. VM uses an approximation to the Least Recently Used (LRU) algorithm in selecting a page to be removed from main memory to make a page frame available.

When a frame is needed, the core table scan mechanism is used to get one from the core table. The core table scan mechanism usually provides a frame belonging to an out-of-queue user. If this is not possible, a frame belonging to an in-queue user must be taken. This is called a "page steal".

The core table scan consists of a pointer steadily advancing through a map of main storage, the core table. In its path it resets the reference

---

[1] A Page Frame is a 4K area of real storage that can be used to hold a page of a user's virtual memory.

[2] The Core Table is a table with one slot per page frame which is used to hold status indicators associated with the frame.

[3] The multiprogramming level is the number of in-queue users, i.e. those in Q1 and Q2 (including Q3), shown as Q1+Q2.

bit(s) associated with a frame, if it was set; if it was reset, it moves the frame onto the free list, thereby making the frame available for other use. The effect of the core table scan is that in-queue users, who reference their pages, tend to maintain their frames, while out of queue users tend to lose their frames. Because of the random distribution of pages in the core table, a user tends to lose only one page at one time. Since the free list requires replenishing whenever it falls below its threshold, pages are usually moved onto the free list one at a time. If the page moved onto the free list is marked changed, then it has to be written out onto a paging DASD, <u>before</u> the frame is actually moved onto the free list.

At queue drop the reference bits of pages belonging to a user are reset, thereby making the frames available for core scan. In-queue users find that it takes one full core table scan to reset all the reference bits associated with their (usually recently referenced) pages, and only a second scan takes the page away, resulting in an average page life of 1.5 scans. All the reference bits are reset at Q-drop time, thus the pages of such a user have a page life of up to one scan (average life 0.5 scan). Thus, the LRU algorithm is strongly biased towards in-queue users. The fact that out of queue users are allowed to hold pages for a while (depending on the speed of core scan) and so the frames remain in main storage, results effectively in "logical swapping" the user. If the user requests another service before core scan had a chance to reuse the user's frames, the pages are found in main storage and used when referenced. The totality of frames belonging to users dropped from queue is called the paging buffer, and specifically the pages belonging to former Q1 users is called the <u>Q1 or interactive buffer</u>.

PAGE ALLOCATION AND CYLINDER SELECTION

In VM the device for a page-out allocation is selected on a basically <u>round robin</u> basis, i.e. each device is selected in turn. The paging algorithm remembers the last cylinder used on a paging device and it attempts to find an unused slot for a page-out on that cylinder.

If it cannot do so, it locates the available slot nearest to the center of the volume. If the center of the volume is within the paging area, then the pages will be clustered around that point. The intention of this <u>zig-zag</u> search is to reduce seek distances on the device. If the center is outside the page data set, then the pages will be packed at the edge nearest to the center.

SHORTCOMINGS OF VM PAGING

Because of the single paging requests large systems using VM usually required low access time paging devices and the available CPU power was sometimes not realizable due to real storage constraints.

Scheduling and paging were not directly related, i.e. main storage was not preferentially allocated to otherwise preferentially treated work.

On large systems management of main storage has manifested itself as a problem in some cases.

The zig-zag allocation algorithm presents problems on large systems since steadily increasing seek distances may result.

THE IMPORTANCE OF RESIDENT PAGES

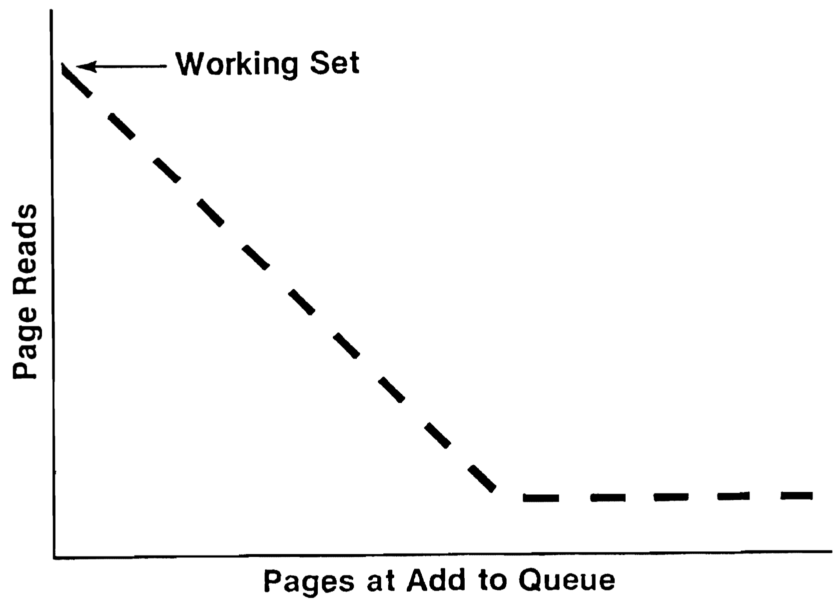

Figure 2. The importance of resident pages

It is known that the more main storage that exists in a VM system, the less paging that takes place. One may assume that this occurs because fewer pages are stolen from active users. In fact, what happens is that the pages belonging to Q-dropped users are retained in main storage and are reused when these users are Q-added. Figure 2 demonstrates this phenomenon. Traces of user activity indicated that the more frames a CMS user still owns in main storage at Q-add time the fewer page faults he suffers.

One can therefore conclude that CMS users have a working set that crosses "transactions", i.e. Q-stays (not CMS transactions). From this we may infer that explicit pre-paging might work, i.e. fetching the pages belonging to a user's working set all at once would be beneficial. This is the basis of the swapping concept in VM/SP HPO 3.4

VM/SP HPO 3.4 PAGING ENHANCEMENT OBJECTIVES

The basic objective for the VM/SP HPO 3.4 paging enhancements was to improve interactive (CMS) response time by reducing page waits and making the paging subsystem more efficient, especially for large systems. In addition, blocking techniques were used to reduce the number of times paging paths were used.

Block paging: Understanding that the use of single page SIOs is not very efficient for DASD or for the CPU, an attempt is made to use "block paging" when possible. Thus, two kinds of block paging are introduced. Swapping is used to move user working set pages into and out of main storage. Page-out operations are also blocked. Block paging exploits the high data transfer rate capability of DASD. It effectively establishes "big pages", transferring multiple related pages with a single SIO. It is equivalent to transferring a single big page. Note that a granularity advantage is obtained since the small page components of this big page can be (are) changed with time, which would not be true for a single big page. The big page concept reduces seeks and latencies; at most one seek and one latency are required per SIO, i.e. for each group of multiple pages. Consequently the need for zero seek and minimal latency devices is reduced and often eliminated.

Relate paging to scheduling: In VM systems, frames belonging to inactive users are retained in main storage with no regard whether this user is likely to be dispatched or the page is likely to be used. VM/SP HPO 3.4 attempts to change this by more closely relating real storage use to scheduling.

Identifying and protecting real working sets: In VM the size of a user's "working set" was indirectly estimated in a way that was dependent on the workload in many cases. VM/SP HPO 3.4 attempts to reduce this workload dependency by determining the "real working set"; that is, it tries to identify the actual working set pages and count them to obtain the actual working set size. Furthermore, if main storage is over committed, VM/SP HPO 3.4 attempts to keep in storage only the "needed" working set pages for scheduled users. Specifically, the number of in-queue users is controlled on the basis of storage use and over commitment of main storage is reduced.

Protecting the interactive buffer:. VM/SP HPO 3.4 attempts to improve interactive response time by explicitly maintaining the Interactive or Q1 Buffer, by logically swapping and protecting Q1 users' working sets if possible.

ADVANTAGES OF BLOCKED I/O

```
ONE PAGE PER SIO

|<——A——>|B|  |<——A——>|B|  |<——A——>|B|


TEN PAGES PER SIO

|<——A——>|B|B|B|B|B|B|B|B|B|B|


A = control unit protocol,  seek,
    latency, RPS miss

B = data transfer time for one page
```

Figure 3. Advantages of blocked I/O

Figure 3 illustrates the conceptual advantages of blocked DASD I/O. Each SIO operation incurs an "overhead", "A", consisting of control unit protocol time, seek to the cylinder sought, latency to reach the page sought and potentially an RPS miss if the path is busy when attempting to reconnect to the path to do the data transfer. Thus, in case of a single page being transferred by one SIO, there is the overhead "A" associated with each page transfer "B". In contrast, when multiple consecutive pages are transferred with a single SIO operation, there is a single overhead "A" associated with multiple "B"s. Clearly, the importance of the overhead "A" diminishes.

	Demand Paging	Block Paging
Pages per SIO	1	10
Actuators per path	4	2
Paging rate per path	60	200
Paging rate per actuator	15	100
SIO response time	29 ms	48 ms
Average time for one page	29 ms	4.8ms

Figure 4. 3380 paging performance in demand and block paging mode

Figure 4 shows the results of a modeling comparison for demand and block paging use of 3380s. The left column indicates that when 3380s are used in demand paging mode, a paging response time of 29 ms is obtained with 15 pages per second per actuator, and four actuators per path. The right column indicates that with block paging (10 pages per SIO) 100 pages per second per actuator per path can be supported with only 2 actuators per path and the SIO completion takes 48 ms, i.e. 4.8 ms per page. Clearly, blocking is advantageous.

PAGE FLOW FOR SWAPPING

Figure 5 on page 13 shows a conceptual image of paging in VM/SP HPO 3.4. The left side of the diagram is very similar to the demand paging subsystem in VM, illustrated on Figure 1 on page 5. When a page-fault occurs and the page resides on a paging DASD, a frame is obtained from the free list, then a page-in operation for one page takes place in the normal fashion into this frame. Page-outs are performed in a similar fashion but now page-out operations tend to be blocked, in contrast to VM.

The right hand side of the diagram represents the new swapping operations. When a user reaches the end of its Q-slice and is Q-dropped, the user's "true" working set is determined. The "true" working set is defined as the non-shared pages which were referenced in the user's Q-stay. The reference bits of these pages are reset. The unreferenced pages are "trimmed", i.e. they are moved onto the flush list from which they will be flushed. The flush list is used explicitly in VM/SP HPO 3.4 for this purpose. Flushing is a normal, expected operation. The user's working set pages are "logically swapped", i.e. his page tables are invalidated, and the user is placed on the swap list. Under normal circumstances the user's (especially an interactive user's) working set pages are retained in main storage,

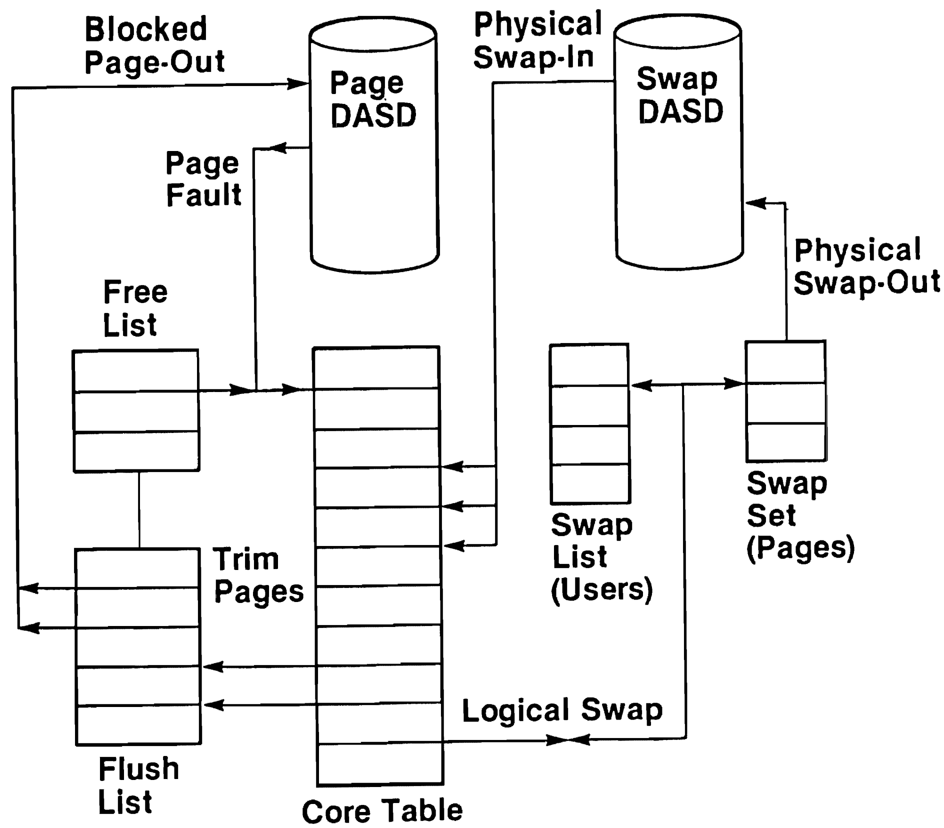

Figure 5. VM/SP HPO 3.4 Paging Schema

even though the page tables are invalidated. The totality of such working set pages belonging to interactive users represents the Interactive Buffer.

Obviously it is not possible to maintain the working set of all users in main storage forever. When a page frame shortage is detected the honeymoon of a logically swapped user comes to an end. Physical swapping takes

---

[4] A swap set is a collection of pages belonging to a user that is written out contiguously into a swap area, i.e. physically swapped out. Subsequently, they are sequentially read back into non-adjacent frames of main storage as a swap-in operation. The size of a swap set is a parameter to be defined at SYSGEN time.

place, releasing some of his frames for other use. In other words, physical swapping is used to replenish the free list.

Physical swap-out means that the next user on the swap list is identified and one or more of his swap sets[4] is swapped out. Users are placed on the swap list in FIFO sequence in the order of their Q-drops. However, there is a separate swap list for Q2 and Q1 dropped users, and Q2 users are physically swapped out before Q1 users. This provides preference to Q1 users.

This swapping mechanism provides at least two blocking advantages. Since all the pages written by a single SIO are contiguous, device busy time is reduced. Also, the blocking means that the number of SIOs and interrupts is reduced, and thus many CPU cycles are saved.

For a physical swap a user's working set pages are organized into swap sets. The working set of a user may require several swap sets. When a physical swap is necessary, one or several swap sets, but not necessarily all swap sets belonging to a user, will be written to the swap data sets. Thus, a user can be partially swapped. If this is the case, the next physical swap-out operation forces another swap-out for this user, since all the logically swapped pages of a user are swapped out before another user is selected from the swap list as a candidate for physical swapping.

When a user is Q-added, this normally causes a page-fault operation to take place, usually to the first page of in the user's address space. If the page fault occurs to a page in a swap set, then the entire swap set is swapped in. This is called "swap-faulting", which again provides the blocking advantages described earlier. A swap-fault always causes the swap-in of that swap set. It can possibly cause the swapping of more than one swap set, but it does not (necessarily) cause swap-in of all the swap sets belonging to the user. If all the swap sets were always swapped in or out at once, a large user (e.g. an MVS guest) could easily overwhelm the paging subsystem. After a swap-in operation the referenced bits of the pages swapped in are reset in order to avoid misleading reference indications.

In VM, Q1 working sets tend to remain close to a median size (e.g. 24 frames) in any given installation. If the installation has multiple swap devices on independent paths, Q1 response time can be reduced by "pre-paging" multiple swap sets. If the installation has a pre-page value of "n", VM will initiate a physical page in of "n" swap sets when the user is added to queue. If these swap sets reside on multiple devices accessible on distinct paths, these swap sets will be transferred concurrently. Consequently swap response time, and therefore user response time, will not include the time to transfer "n" swap sets; rather it will only include the time to transfer 1 swap set. For example, if:

- A median user size is 24 frames
- The swap set size is 8 frames

then maximum concurrency can be obtained using:

- A pre-page size of 3 (24 frames)
- At least 3 separate swap devices
- Each swap device on an independent path.

In VM/SP HPO 3.4 physical swap-out, not the core table scan as in VM, is the major source of frames for the free list. If for some reason swap-outs do not provide enough frames to replenish the free list then user page "stealing" (via core table scan) remains as the mechanism of last resort. If core table scan is invoked frequently, as demonstrated by a high steal rate in VMAP, then the swap mechanism is not functioning properly. In fact, a steal rate exceeding one percent should be cause for investigation.

Since most of the time swap-in operations require more than one frame at a time, the free list had to be made larger than before.

SWAP SET AFFINITIES

At Q-drop VM/SP HPO 3.4 identifies all referenced pages during the previous Q-stay. These can be considered the real working set pages (5) since the Q-slice is normally very small. It is immaterial whether these referenced pages are changed or unchanged, by our definition these are "needed pages". These pages are logically swapped; the "unneeded" pages are trimmed to the flush list.

When physical swapping takes place, swap sets are formed by grouping the logically swapped pages of a given user in order of <u>virtual addresses</u> into sets of pages. The number of pages in a swap set is a SYSGEN settable parameter.

Thus, swap sets are related by virtual address and by <u>time of reference</u>, since all the pages in a swap set were referenced during the same Q-stay. The collection of pages forming a swap set is constructed for a swap-out, and the swap set exists on DASD until it is swapped in, after which the affinity of pages is lost. After each new Q-stay, swap sets are formed afresh for swap-out and the pages in them may be different than before. Note that not all swap sets of a virtual machine come necessarily from the same queue stay. If a swap set is not brought into main storage during a Q-stay, its identity can be retained across multiple Q-stays. In most instances however, especially for interactive users a swap set exists only between consecutive Q-stays. Swap sets are really BIG pages, since they are always written and read with a single SIO onto consecutive slots on DASD, and a page fault to any one, a" swap fault", causes the entire swap set to be read.

PAGE-OUT DEVICE AND SLOT SELECTION

The page allocation algorithm to paging areas is changed in VM/SP HPO 3.4, in order to encourage the frequency of block page-outs. A new device selection algorithm is used; "N-select" replaces the VM "round robin" selection algorithm. The basic concept of N-select is to select the same page-out device (up to) 8 times, unless a cylinder boundary is reached. The repeated selection of the same device allows grouping multiple page-out requests within the same SIO operation, thereby blocking page-out requests. Page reads however tend to remain unblocked. The result of page-out blocking is a higher chain percentage reported in VMAP.

The zig-zag slot selection algorithm of VM is replaced with a "Moving cursor" algorithm. The moving cursor algorithm maintains a pointer that moves across the paging area. The cursor points at a slot which is to be used for allocation if the slot is free. The cursor is advanced steadily across the paging area as required by the allocations taking place. The expectation is to find empty, unused areas ahead of the cursor. This construction was devised to provide a high probability for the existence of contiguous empty slots, which in turn provides effective blocked page-outs. Also, seek frequency and distance decrease within a device. Experience indicates that as the allocation cursor is advanced, a wave of page references moves across the DASD volume. The allocation writes occur ahead of the cursor and page reads occur behind the cursor. But the band of pages referenced frequently is rather narrow, i.e. there is a well-defined locality of reference. The cursor is reflected at the end of the paging area to avoid long seeks which would be necessary if allocation were then begun at the beginning of the area.

The combination of N-select and moving cursor algorithms tends to result in an effective page-out blocking method. Care must be used that the paging area is large enough to allow contiguous page allocation but is not too large so that seeking does not become excessive. As a rule of thumb, paging areas should be defined about six times as large as the maximum number of slots used.

THE SYSPAG MACRO

In VM/SP HPO 3.4 the SYSORD macro of VM is replaced by a new macro, SYSPAG, which is similar to SYSORD in external format. The SYSPAG macro is used to define how the CP-owned volumes are to be allocated and used. This new macro provides for the definition of the entire paging hierarchy, and specifically for the definition of a new type of area to be defined on DASD for the swapping function, referred to as "swap areas"; it also allows definition of swap set sizes. SYSPAG gives an installation greater degree of control over the specification of the system paging and spooling areas. The effect of the SYSPAG macro (shown in Figure 6) is to combine the definition of CKD and FBA devices into similar tables to be handled by the same set of instructions.

```
SYSPAG (volid[,[1st cylinder][,last cylinder]],
       TYPE=xx[,SWSIZE=nn])

Some (area) types are:
 SW - swapping
 PP - preferred paging
 PG - general paging
 PM - page migration
 PS - spooling
 DU - dump spool files

SWSIZE defines number of pages
in a swap set.
```

Figure 6. The SYSPAG macro

Paging areas are defined in the SYSPAG macro by specifying the volume serial, the first and last cylinder of the paging areas and the paging area type. A single swap set size is defined for the whole system.

MULTIPROGRAMMING LEVEL CONTROL

In VM users were Q-added if the projected working set of the user plus the sum of the working sets of in-queue users (SUMWSS) was less than the Available Frame Count (APAGES):

Projected Working Set Size + SUMWSS < APAGES

Q1 users were given preference because, APAGES was increased by a factor of 1.25 for Q1 users, thus Q1 users rarely noticed the reality of main storage constraint. The above calculation ignores the requirement for an interactive buffer. Many installations attempted to obtain that by artificially lowering APAGES. VM/SP HPO 3.4 explicitly provides for the existence of the Interactive Buffer, thus APAGES is reduced by the Interactive Buffer size (IBUF), the number of frames required by Q-dropped I/A users in the last N seconds. Thus, <u>non-interactive users</u> are Q-added only if the more stringent criterion is met:

Projected Working Set Size + SUMWSS + IBUF < APAGES

For interactive users, preference is provided by using the formula without IBUF. The rationale is that an interactive buffer should not be kept at the expense of newly added interactive users. The effect of these modifications is to reduce excessive use of main storage by non-interactive users. Furthermore, VM permitted Q2 to Q1 transitions, so that non-interactive users could masquerade as interactive. This is ro longer permitted.

In summary, the interactive buffer is explicitly taken into account before Q-adding non-interactive users. This effectively reduces the number of non-interactive users in main storage when compared with VM.

SET COMMANDS

A variety of new SET commands have been introduced for more effective system control.

SET MINWS Command

A default system-wide minimum working set size of two swap sets is established for interactive users. This means that if the user has at least two swap sets worth of pages at Q-drop time, his working set will not be trimmed below that value. In addition, at least this much will be physically swapped out, if physical swap takes place. Normally non-interactive users have many more pages than two swap sets worth, (i.e. 20 pages if a swap set size of 10 is used). The effect of this value is to guarantee less trimming for interactive users. This translates into more physical swapping and less demand paging. The SET MINWS command allows changing this value system-wide or for a particular user. When used, the number of frames set should be set as an integral multiple of the swap set size. This command can be used in three ways:

1. For seldom used service machines, a "protected working set" can be established <u>for that user</u> so that the service machine can be freely swapped instead of being page-faulted.

2. If the installation determines the high median size of its interactive users (e.g. 30 frames), then this parameter can be set <u>system-wide</u>.

3. If the installation determines the high median size of a subset of its interactive users (e.g. 30 frames), then this parameter can be set one at a time for all these users.

SET SRM IBUFF Command

The interactive buffer consists of the logically swapped working sets of interactive users "recently" dropped from queue. The VM/SP HPO 3.4 system provides for a default setting of "recently" as 10 seconds. VM/SP HPO 3.4 also provides for a default maximum interactive buffer size of 50 % of the pageable storage. Both of these parameters can be changed by the SET SRM IBUFF command. Since interactive response time is often determined by the size of the interactive buffer, the installation can use this control to increase batch throughput at the expense of interactive response time or vice versa.

SET SRM PREPAGE Command

In order to take advantage of the parallelism of swap paths the VM/SP HPO 3.4 system provides for a default swap-in size of 2 swap sets for interactive users. This means that when an interactive user is Q-added, two of his swap sets will be "pre-paged", i.e. swapped in if there are at least 2. This reduces interactive response time. The default is NO pre-paging for non-interactive users. The command allows these defaults to be changed, separately for Q1 and Q2 users. The installation may increase the Q1 value if there are sufficient swap paths and the median size of interactive users is greater than 2 swap sets. In installations where non-interactive users predominate, the Q2 default may be increased.

SET SRM SWPQTIME Command

In order to prevent users from occupying main storage unnecessarily, the system must ensure that users do not stay logically swapped forever when there is a demand for main storage. A main storage constraint is relieved by physical swapping. Physical swapping is performed from the logical swap queues. There are two such queues, one for interactive users, one for non-interactive users. Users are physically swapped from the non-interactive queue first. VM/SP HPO 3.4 establishes a 20 second default time for interactive users staying on the interactive logical swap queue. After this time the interactive user is moved onto the non-interactive logical swap queue, where it competes against non-interactive users. The default for maximum time on the non-interactive queue is set as 100 seconds. Both these default values can be changed by the SET command.

Set SRM MINNUMSS Command

Most of the paging in VM/SP HPO 3.4 occurs as swap paging. Therefore the requirement for free frames from the free list increases, since each swap set read requires a swap set worth of frames (usually 8 or 10 frames). Consequently the size of the free list is increased from:

Q1 + Q2 + 1 in VM

to:

(SS * M) + (Q1 + Q2 +1) in VM/SP HPO 3.4

where: SS = the size of the swap sets (in frames)
M = a number set by the SET SRM MINNUMSS command (default is 6)

This parameter is not normally changed by an installation unless it can be observed in VMAP that the free list is frequently depleted.

PAGING CONTROL OF NON-INTERACTIVE VIRTUAL MACHINES

The category of service and guest machines often presents a special problem. These machines are not necessarily interactive themselves, and therefore they are classified in Q2, but their responsiveness is very important. This responsiveness cannot be obtained if these machines suffer frequent page faults. In VM, the SET RESERVE command allowed the installation to "reserve" a number of frames for one virtual machine. This guaranteed a number of frames to be retained in main storage for this virtual machine. VM/SP HPO 3.4 extended this concept to multiple virtual machines. Furthermore, the algorithm for identifying the frames has been improved. In VM the first such frames referenced were retained. In VM/SP HPO 3.4 the normal referencing pattern of the service machine identifies the frames to be retained at each Q-drop with the provision that at least the reserved number of pages are retained in main storage if they exist for the virtual machine. Thus, an LRU algorithm is executed for the reserved machine, and the machine's real working set is protected.

Alternatively, it is possible to enhance the performance of these machines by the use of the

SET SRM PREPAGE Q2

command, which allows for pre-paging of swappable non-interactive machines, probably in combination with the

SET MINWS virtual-machine-id

command, which explicitly defines the working set size for this machine.

DISPOSABLE PAGE COLLECTION

Pages belonging to virtual machines are handled by logical swap, trim and physical swap. There are pages not belonging in this category, such as:

Shared segment pages
CP pages
User pages not in working set
(obtained without Q-add and not trimmed)
Spool buffer pages

In VM the "core table scan" disposed of these pages. In VM/SP HPO 3.4 core table scan is not normally invoked (except as a last resort), thus a new mechanism is required to handle "disposable" pages of this type. Thus a periodic core table scan is introduced which looks for these pages. It scans 1/8th of the dynamic paging area every 4 seconds, resets the reference bits of pages of this type, and disposes of unreferenced "non-user" pages by putting them on the flush list, i.e. effectively establishing a working set for this type of page and trimming the disposable ones. Clearly all such pages are examined in 32 seconds, and the unreferenced page lifetime is 48 seconds on average for these kind of pages. There is no SET command to change these constants.

ADDITIONAL MEASUREMENT SUPPORT IN VM/SP HPO 3.4

The new facilities provided in VM/SP HPO 3.4 require additional measurement facilities. Thus, new VM monitor records are provided:

- On use of each swapping and demand paging areas and devices.
- On each logical swap-out and swap-in.

THIS PAGE INTENTIONALLY LEFT BLANK

Term	Page Discussed
Swapping	
Logical swap	7
Physical swap	13
Swap set	13
Trim pages	13
True working set	12
Swap fault	15
Swap area	13
Pre-paging	14
Logical swap queue	19
Scheduler	
Interactive Buffer	7
Paging	
Demand paging	3
Block page-out	10
N-select	15
Moving cursor	16
Disposable page collection	21

Figure 7. Glossary

A whole vocabulary of new terms has been introduced by the VM/SP HPO 3.4 Paging Enhancements. A list of these new terms is given in Figure 7.

THIS PAGE INTENTIONALLY LEFT BLANK

MODELING COMPARISON OF PAGING AND SWAPPING DASD

Figure 4 on page 12 presented a performance comparison of 3380 demand paging and swapping devices. This comparison is based on modeling data which is presented below.

```
1  PAGE  per SIO is assumed

SIO RATE PER PATH:                                = 60  SIOs
CONTROL UNIT OVERHEAD                             =  2.6ms
DATA TRANSFER TIME (1 page)                       =  1.5ms
AVERAGE CHANNEL SERVICE TIME: 2.6 + 1.5           =  4.1ms
PATH UTILIZATION    = 60 x 4.1 = 246 ms/s         = 24.6%
PATH  UTILIZATION PER ACTUATOR = .246 / 4         =  6.1%
RELATIVE PATH (CHANNEL) UTILIZATION: RCB
  3 x .061 / (1-.061) = 0.195                     = 20  %

COMMAND DELAY, CD     = 4.1 x .20 / 2             =  0.4ms
OVERHEAD TIME                                     =  2.6ms
SEEK TIME                                         =  6.3ms

RPS MISS = 16.7 x .2 / (1-.2 )                    =  4.2ms
LATENCY TIME                                      =  8.3ms
DATA TRANSFER                                     =  1.5ms
DEVICE SERVICE TIME                               = 22.9ms

DEVICE UTILIZATION      = 15 x 22.9 =  .34        = 34  %

QUEUE WAIT = (22.9 x .34) / (2 x (1-.34))         =  5.9ms
DASD TIME: 22.9 + 5.9 + .4                        = 29.2ms
```

Figure 8. Model of four 3380 Demand Paging Actuators

Beretvas (6) provides a method for calculating paging response times. This method is used in Figure 8. A configuration of a dedicated channel with four 3380 paging actuators is assumed. An M/D/1 model is used. A paging load of 60 pages per second per path, one page per SIO, is assumed, evenly distributed among the four actuators.

The channel busy time associated with each SIO, the average channel service time, is the sum of the data transfer time, 1.5 ms and the protocol, 2.6 ms, for a total of 4.1 ms. The total path utilization is calculated as the SIO rate multiplied by the path busy time per SIO, 60 times 4.1 yields a path utilization of 24.6%. The path utilization per actuator is one fourth of this. (6.1 %) To calculate the RPS miss time later, the relative path (channel) busy percentage has to be calculated. Relative path busy represents the effective path busy as seen by a device when it tries to reconnect to the path. This calculation must clearly exclude the times when the device itself is busy. Thus, the total path busy as seen by any device is the path busy caused by the other three devices, i.e. 3 x 0.061. Furthermore, the time period observed must exclude the time when the device itself is busy. Therefore the total time period in the denominator has to be reduced by the fraction of the time during which the path busy was caused by the device itself (0.061). Thus, a relative path utilization of 20% is calculated. (As a first approximation, the actual path utilization of 24.6% could have been used). Channel command initiation delay is calculated as 0.4 ms, by using the average channel service time and the relative channel busy percentage obtained before.

Device service time per page is calculated as 22.9 ms by adding the data transfer time of 1.5 ms, the protocol time of 2.6 ms, the average latency of 8.3 ms, an assumed seek time of 6.3 ms, and a calculated RPS miss time of 4.2 ms. The device utilization of 34% can then be obtained as the product of the paging rate per actuator, 15 pages per second, and the device service time of 22.9 ms.

Queueing delay of 5.9 ms is obtained using an M/D/1 formula from the device service time and device utilization. The page-in response time, 29.2 ms is obtained as the sum of the device service time, the queueing delay and the command delay.

The modeling technique presented can be used for any of the paging device types. Curves can be plotted indicating expected page-in times as a function of paging rates.

In Figure 9 on page 27 a configuration of a dedicated channel with two 3380 swap actuators is assumed. An M/D/1 model is used. A paging load of 200 pages per second per path, ten pages per SIO, is assumed, evenly distributed between the two actuators.

The data transfer time for the transfer of 10 pages is the duration one revolution, 16.7 ms. The channel busy time associated with each SIO, (the average channel service time) is the sum of the data transfer time (16.7 ms) and the protocol time (2.6 ms) for a total of 19.3 ms. The total path utilization is calculated as the SIO rate multiplied by the path busy time per SIO. This is 20 times 19.3, yielding a path utilization of 38.6%. To calculate the RPS miss time later, the relative path (channel) busy has to be calculated. The total path busy as seen by a device is the path busy caused by the other device, i.e. 0.193. Furthermore, the time period observed must exclude the time when the device itself is busy, thus the total time in the denominator has to be reduced by the fraction of the time the path busy is caused by the device itself (0.193). A relative path utilization of 24% is calculated. Channel command initiation delay is calculated as 2.3 ms by using the average channel service time and the relative channel busy percentage obtained before.

Device service time per SIO is calculated as 35.9 ms by adding the data transfer time of 16.7 ms, the protocol time of 2.6 ms, the average latency of 8.3 ms, an assumed seek time of 3 ms, and a calculated RPS miss time of 5.3 ms. The device utilization of 36% can then be obtained as the product

```
10  PAGES  per SIO are assumed

SIO RATE PER PATH:                              = 20 SIOs
CONTROL UNIT OVERHEAD per SIO                   =  2.6ms
DATA TRANSFER TIME (10 pages)                   = 16.7ms
AVERAGE CHANNEL SERVICE TIME: 2.6 + 16.7        = 19.3ms
PATH  UTILIZATION    = 20 x 19.3 = 386 ms/s     = 38.6%
PATH UTILIZATION PER ACTUATOR = .386 / 2        = 19.3%
RELATIVE PATH (CHANNEL) UTILIZATION: RCB
  .193 / (1-.193)  = 0.239                      = 24  %

COMMAND DELAY, CD  = 19.3 x.24 / 2              =  2.3ms
OVERHEAD TIME                                   =  2.6ms
SEEK TIME                                       =  3.0ms

RPS MISS = 16.7 x .24 / (1-.24)                 =  5.3ms
LATENCY TIME                                    =  8.3ms
DATA TRANSFER                                   = 16.7ms
DEVICE SERVICE TIME                             = 35.9ms

DEVICE UTILIZATION = 10 x 35.9 =  .36           = 36  %

QUEUE WAIT = (35.9 x .36) / (2 x (1 -.36))      = 10.1ms
DASD TIME: 35.9 + 10.1 + 2.3                    = 48.3ms
```

Figure 9. Model of two 3380 Swap Actuators

of the SIO rate per actuator, 10 SIOs per second, and the device service time of 35.9 ms.

Queueing delay of 10.1 ms is obtained using an M/D/1 formula from the device service time and device utilization. The swap-in DASD response time of 48.3 ms is obtained as the sum of the device service time, the queueing delay and the command delay.

THIS PAGE INTENTIONALLY LEFT BLANK

The experiments with the prototype provided some useful observations

1. Demand Swap (swap fault) works. There is indication that it is worthwhile not swapping in all the swap sets at the same time, since occasionally there are swap sets that remain unused. It was beneficial to limit the automatic pre-paging at the beginning of a transaction. It limits the loading effect on the swapping subsystem that would be caused by the immediate pre-paging of a large number of swap sets. Further, the potential storage pressure is also reduced.

2. Page Steal Rate is a sensitive indicator. When free list value was not set high enough, significant (higher than one percent) stealing occurred. Clearly in a swapping system stealing is undesirable.

3. Multiprogramming Level control is important. When the prototype was initially run without explicit MPL control, too many Q2 users were Q-added. Thus, the prototype was changed to run with a "fixed" Q2 MPL control. Furthermore, the product was designed to provide dynamic MPL control using the interactive buffer concept.

4. Overall paging rate increases but demand paging rate is much lower — most of the paging is due to swapping. The primary reasons for the increased paging rate are:

   - Increased throughput requires more paging.
   - Some of the pages swapped in are not really required.
   - Pre-paging causes pages to occupy frames before they are actually referenced. This may increase page residency time.

   However, in all observed cases, the demand paging rate was dramatically decreased, thereby reducing (or eliminating) the need for 2305s.

5. 3380 devices are excellent for swap paging. In the 3081D measurements a paging rate of up to 800 pages per second was easily supported by 3380s. Since for swapping load the high data rate dominates, 3380s are eminently suitable devices.

6. Pre-paging large (Q2) users may be destructive. In an early version of the prototype the entire working set was swapped and the 3350s used could not support the heavy instantaneous paging load that resulted. Also, total swapping created an instantaneous requirement for a large number of free page frames. Thus, swap faults were invented, and pre-paging control was introduced.

7. The old size of the free list is inadequate in a swap environment. Pre-paging requires many free pages suddenly and the old free list could not supply frames fast enough. Accordingly a SET command con-

trol was introduced to regulate the free list size, and the default was set higher than before.

8. Blocked page-outs are helpful. An early version of the prototype has shown significant improvements just by blocking page-outs and no other change.

1. Who should go to VM/SP HPO 3.4?

   In our opinion, large systems, i.e. systems of 4341-2 size or larger (especially those with response time problems) who have adequate channel capacity for the generated swap load.

2. Can I install VM/SP HPO 3.4 without reconfiguring the system?

   You must reconfigure using SYSPAG rather than SYSORD but you can retain the old demand paging configuration. For improved performance obtainable from blocking it is advisable, however, to reconfigure the paging DASD configuration so that swap areas are also defined.

3. Where will I get configuration help?

   A Washington Systems Center Technical Bulletin will be published on the basis of early installation experiences.

4. Can you tell me how much increase can I expect in the paging rate as a result of swapping?

   The results depend on your particular configuration, but as a rough rule of thumb, you can expect an increase of 20-50 percent.

5. How do I set up my paging configuration?

   There are some basic considerations, some of which are listed below.

   a. The fastest paging devices should be used for swapping.

      Thus, 3380s are eminently suitable for swapping. If the installation has a limited number of 3380 actuators, they should be preferably used as swap rather than paging devices.

   b. Distribute swapping onto as many paths as possible.

      Since each swap device keeps a path busy a significant portion of the time, it is wise to have only <u>one</u> swap device per path. It is possible to have a swap device and one or more paging device per path.

6. Can you give me an easy to use guideline for configuring my paging subsystem?

   Beretvas (6) provides a methodology for configuring the system. The guidelines below summarize these techniques.

   a. Observe your current VM maximum paging rate.

   b. Increase it by 30 percent to allow for swapping.

c. Assume that 75 % of the total paging rate goes to swap areas, 25 % to page areas. Determine the number of paging actuators required using the table given below. Configure the system using one swap actuator per path, and a maximum of four paging actuators per path. Use a swap set size of 10 in a system with 3380s and a swap set size of 8 for all other cases.

Recommended Maximum Paging Rates Per Actuator

	Demand paging	Swap paging
3330	15	30
3350	17	50
3375	20	75
3380	25	100
3880-11	60	

d. As an example, assume a VM system with a paging rate of 306 pages per second. With VM/SP HPO 3.4 this rate can be expected to grow to 400 pages per second-100 pages per second to page areas and 300 pages per second to swap areas. Using 3380s it can be seen from the above table that three swap and four paging actuators are required. The configuration can be set up by spreading one swap and one paging actuator per path, i.e. a configuration of (SP,SP,SP,SP) would suffice, giving four paths for swapping and paging. Note that the paths are mixed between paging and swapping.

7. Can I use my 3370s (or 3880-11s) for swapping?

FBA devices and 3880-11 are not supported for swap areas. They can be used as demand paging areas only. In addition, 2305s are not good candidates for swap devices. Their advantages are zero seek and short latency times. Data transfer is the most important component in a swap operation, thus making the 3380 the ideal device.

8. How large should I make my preferred paging areas?

If the paging areas are made too large, excessive seeking may result; if they are made too small, then contiguous page-out allocations will not occur. Count the number of slots you require in a paging device, and multiply it by three to obtain the size of the paging area required.

9. How large should I make my swap areas?

If the swap areas are made too large, excessive seeking may result; if they are made too small, then overflow to the paging areas will not occur. Count the number of logged-on users, multiply it by the average user size (in frames) then multiply the product by two to obtain the size of the swap area.

E.g. Take 400 users with an average size of 25 slots. The requirement is then (400 x 25 x 2) = 20000 slots, about 80 MB.

10. If I initially configure and run without swap areas, what should I do when I decide to use swap areas?

   a. Use VMAP to determine the trim (demand) page write rate and the working set write rates for peak load.

   b. Configure for paging by determining the number of paging actuators required for twice the trim page write rate.

   c. Configure for swapping by determining the number of paging actuators required for twice the working set page write rate.

THIS PAGE INTENTIONALLY LEFT BLANK

The VM/SP HPO 3.4 paging enhancements improve interactive response time by improving the efficiency of the paging subsystem. The efficiency of the paging subsystem was improved in several ways. The system now makes better use of real memory and thus makes some demand paging unnecessary. In many cases multiple contiguous pages are transferred following one DASD access, thus reducing average access time and DASD busy time.

VM/SP HPO 3.4 is only an adjunct to large real memory, and is especially useful for large systems.

VM/SP HPO 3.4 provides the most benefit in heavy paging environments.

VM/SP HPO 3.4 provides additional performance controls for service machines.

Use of VM/SP HPO 3.4 may be difficult on small systems with limited paging I/O capabilities.

The VM/SP HPO 3.4 paging enhancements improve interactive response time and also improve throughput because the CPU supervisor time is reduced. The page blocking techniques of VM/SP HPO 3.4 reduce the CPU supervisor time because of the reduction of SIOs and I/O interrupts.

THIS PAGE INTENTIONALLY LEFT BLANK

1. W. J. Doherty and R. P. Kelisky: "Managing VM/CMS systems for user effectiveness," IBM Systems Journal, 18, No. 1, (1979).

2. A. J. Thadhani: "Interactive User Productivity," IBM System Journal, 20, No 4. 407-423 (1981).

3. W. Tetzlaff, T. Beretvas: "A New Approach to VM Performance Analysis", SHARE 58.

4. T. Beretvas and W. Tetzlaff: "Paging in VM/SP", Washington System Center Technical Bulletin, GG22-9319, May 1983

5. P.J. Denning: "The Working Set Model for Program Behavior", CACM 11, 323-333 (May 1968).

6. T. Beretvas: "Page/Swap Configurations", G320-0053, IBM Corporation Marketing Publications, Department 824, 1133 Westchester Avenue, White Plains, New York 10604.

7. S. Friesenborg: "DASD Path and Device Contention Considerations," IBM Washington System Center Technical Bulletin, GG22-9217-0, March 1981.

READER'S COMMENT FORM

Title: Paging Enhancements in VM/SP HPO 3.4
Washington Systems Center
Technical Bulletin GG22-9367

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Please state your occupation: ______________________________

Comments:

Please mail to: William O'Brien
Washington Systems Center
18100 Frederick Pike
Gaithersburg, MD 20879

Reader's Comment Form

Cut or Fold Along Line

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

William O'Brien
NAD Washington Systems Center
IBM Corporation
18100 Frederick Pike
Gaithersburg, MD 20879

Fold and tape Please Do Not Staple Fold and tape

IBM®

READER'S COMMENT FORM

Title: Paging Enhancements in VM/SP HPO 3.4
Washington Systems Center
Technical Bulletin GG22-9367

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Please state your occupation: ______________________________

Comments:

Please mail to: William O'Brien
Washington Systems Center
18100 Frederick Pike
Gaithersburg, MD 20879

Reader's Comment Form

Cut or Fold Along Line

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

William O'Brien
NAD Washington Systems Center
IBM Corporation
18100 Frederick Pike
Gaithersburg, MD 20879

Fold and tape | Please Do Not Staple | Fold and tape

READER'S COMMENT FORM

Title: Paging Enhancements in VM/SP HPO 3.4
Washington Systems Center
Technical Bulletin GG22-9367

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Please state your occupation: ____________________________

Comments:

Please mail to: William O'Brien
Washington Systems Center
18100 Frederick Pike
Gaithersburg, MD 20879

Reader's Comment Form

Cut or Fold Along Line

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

William O'Brien
NAD Washington Systems Center
IBM Corporation
18100 Frederick Pike
Gaithersburg, MD 20879

Fold and tape Please Do Not Staple Fold and tape

IBM ®

GG22-9367-00

Paging Enhancements in VM/SP HPO 3.4

Printed in U.S.A. GG22-9367-00