VM/ESA Release 2.1 Performance Report

June 24, 1993

S.P. Lyons

Rt. 17C and Glendale Drive
Endicott, New York 13760

Contents

Notices

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 were determined in various controlled laboratory environments and are for reference purposes only. Customers should not adapt these performance numbers to their own environments as system performance standards. The results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

This publication refers to some specific APAR numbers that have an effect on performance. The APAR numbers included in this report may have prerequisites, corequisites, or fixes in error (PEs). The information included in this report is not a replacement for normal service research.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM′s program may be used. Any functionally equivalent program may be used instead.

Programming Information

This publication is intended to help the customer understand the performance of VM/ESA Release 2.1 on various IBM processors. The information in this publication is not intended as the specification of any programming interfaces that are provided by VM/ESA Release 2.1. See the IBM Programming Announcement for VM/ESA Release 2.1 for more information about what publications are considered to be product documentation.

Trademarks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

ES/9000
ESA/370
IBM
OfficeVision
OfficeVision/VM
PR/SM
Processor Resource/Systems Manager
SAA
VM/ESA
VSE/ESA
VTAM

Acknowledgements

The following people contributed to this report:

Endicott Programming Laboratory

M. S. Bidwell
W. J. Bitner
C. T. Bradley
D. S. Davis
P. A. Duncan
W. G. Ernsberger
G. S. Gasper
W. J. Guzior
G. A. Hine
G. J. Kudamik
S. P. Lyons
S. T. Marcotte
V. E. Meredith
L. L. Quinn
S. E. Shearer
A. M. Shepherd
D. P. Spencer
L. D. Stapleton
C. R. Terry
J. L. Thrall
T. M. Walker

Washington Systems Center

C. L. Morse

Southern Area System Center

G. S. Eheman

Abstract

The *VM/ESA Release 2.1 Performance Report* summarizes the performance evaluation of VM/ESA Release 2.1. Measurements were obtained for the CMS-intensive, OfficeVision*, VSE guest, and VMSES/E environments on various ES/9000* processors.

This report provides performance and tuning information based on the results of the VM/ESA Release 2.1 performance evaluations conducted by the Endicott Programming Laboratory.

Discussion concentrates on the performance effects of migrating from VM/ESA Release 2 to VM/ESA Release 2.1, exploiting the new functions provided in VM/ESA Release 2.1, and using certain tuning options. Some additional evaluations are also included.

Referenced Publications

The following VM/ESA* publications are referred to in this report:

- *VM/ESA: Performance,* SC24-5642
- *VM/ESA: SFS and CRR Planning, Administration, and Operation,* SC24-5649
- *VM/ESA: Installation Guide,* SC24-5526
- *VM/ESA: Planning and Administration,* SC24-5521
- *VM/ESA: Service Guide,* SC24-5527
- *VM/ESA: CP Command and Utility Reference,* SC24-5519

The following IBM High Level Assembler publication is referred to in this report:

- *IBM High Level Asembler for MVS, VM and VSE Presentation Guide,* GG24-3910

The following performance reports are unclassified and may be ordered by customers.

- *VM/ESA Release 1.1 Performance Report,* GG66-3236
- *VM/ESA Release 2 Performance Report,* GG66-3245
- *VM/ESA Release 2.1 Performance Report,* (this report)

The following performance report is classified as IBM* Internal Use Only. The information contained in this document may be discussed with customers, but the document may not be given to customers. They may ask their IBM representative for access to the information contained in this publication.

- *VM/ESA Release 1.0 Performance Report,* ZZ05-0469

Summary of Key Findings

This report summarizes the performance evaluation of VM/ESA Release 2.1. Measurements were obtained for the CMS-intensive, OfficeVision, VSE guest, and VMSES/E environments on various ES/9000 processors. This section summarizes the key findings. For further information on any given topic, refer to the page indicated in parentheses.

Performance Changes: VM/ESA Release 2.1 includes a number of performance enhancements (page 10). The following improvements provided the most benefit to the CMS, office, and VSE guest regression environments:

- improved simulation by CP of LPSWs that load a wait state (all)
- an extension to CCW fast path (guest)
- a 7% reduction in IUCV/APPC path length (CMS, office)
- reduced processing requirements for the EXECUTE XEDIT macro (CMS)

There are a few changes that have the potential to adversely affect performance (page 14). Lastly, a number of changes were made that benefit VM/ESA performance management (page 15).

Migration from VM/ESA Release 2: Benchmark measurements showed the following performance results for VM/ESA Release 2.1 relative to VM/ESA Release 2:

CMS-intensive Internal throughput rate (ITR) improved 1.1% to 1.5%, while external response time ranged from equivalence to a 10% improvement (page 23).

OfficeVision ITR improved 0.7% to 1.4%, while response time improved 5% to 8% (page 44).

VSE guest ITR improved by about 0.6% for the measured V=V cases. With one exception, ITR improved by 2.3% to 2.7% for the measured V=R cases. (pages 54 and 119).

The exception is V=R with full pack minidisks. Because the DASD are not dedicated, this case is not eligible for I/O assist. In VM/ESA Release 2, CCW fast path handled most non-dedicated DASD cases, but not the full pack minidisk case. In VM/ESA Release 2.1, CCW fast path was extended to support this case, resulting in a 21% ITR improvement. These results were obtained using an I/O-intensive workload. Most customer workloads are less I/O-intensive and would therefore experience a correspondingly smaller ITR improvement (page 60).

VMSES/E Overall elapsed time decreased 2%. In addition, VMSES/E now automates checking that formerly had to be done manually. This can potentially result in significant reductions in the total time required to complete a given maintenance task (page 62).

Migration from Other VM Releases: The performance measurement data in this report can be used in conjunction with similar data in the three previous VM/ESA performance reports to get a general understanding of the performance aspects of migrating from earlier VM releases to VM/ESA Release 1.0 (370 Feature) or VM/ESA Release 2.1 (page 68).

New Functions - Virtual Disk in Storage: Several uses of virtual disk in storage were measured. When virtual disks in storage were used for both the VSE lock file and VSE temporary work files, the external throughput rate (ETR) improved by 14% to 29%, while ITR improved by 1.9% to 4.1%. When, in the CMS-intensive environment, virtual disks in storage were substituted for T-disks, the results depended on the amount of paging caused by their real storage requirements. In the case with the smallest paging increase, ITR improved by 0.3%, while external response time improved by 6%. In the case with the largest paging increase, ITR decreased by 1.9%, while response time increased by 23%.

The results show that individual jobs or interactions that make heavy use of virtual disks in storage often experience large elapsed time improvements:

84%	COBOL compile and link job (VSE guest)
50%	COBOL program (CMS)
88%	CMS FORMAT command

Other concurrent work (for which virtual disks in storage do not apply) will generally not be improved and may experience an elapsed time increase as a result of increased system paging (page 76).

Other New Functions: REXX stream I/O was compared to EXECIO and CSL calls—two alternative methods of doing file I/O from a REXX application. Path length data were collected for open, close, read, and write functions. The results indicate that both EXECIO and REXX stream I/O are more efficient than CSL. EXECIO is more efficient than REXX stream I/O for the open/close functions, while the opposite is true for the read/write functions. As a result, REXX stream I/O becomes more efficient than EXECIO when enough records (about 20) are read/written, one at a time, per open/close pair (page 126).

CMS uses a buffer called the “minidisk file cache” to perform read-ahead, write-behind processing for sequentially accessed files. Prior to VM/ESA Release 2.1, it had a fixed maximum size of 8KB. This maximum size now defaults to 8KB, but can be changed (1KB to 96KB) when the CMS nucleus is built. Measurement results suggest that highly storage-constrained systems should stay with the 8KB default but that less constrained systems will tend to benefit from a larger size (page 122).

Additional Evaluations: CMS-intensive measurements taken on the 9121-742 with 1, 2, 3, and 4 processors online show ITR scaling properly as the number of processors is increased (page 128).

Relative ITR results are shown for various cases of VSE/ESA 1.2.0 running on VM/ESA Release 2.1 for the 9121-320 and 9221-170 processors (page 135).

Measurement results compare CTCA to 3745 network connectivity to show how this can affect the performance results (page 137).

CMS-intensive measurements on a 9121-480 showed a 1.2% increase in processing requirements when the CP monitor was running with all sample domains enabled and with default interval and rate settings. The cost of monitoring can vary considerably. Key variables include types of data collected, sample rate, interval rate, processor speed, number of users, user activity, and number of devices (page 142).

Performance can sometimes be significantly improved by placing a suitable restriction on the CP arbiter function. Results from a case study are discussed and suggestions are provided (page 146).

Measurement results comparing the IBM High Level Assembler to Assembler H Version 2 suggest that the High Level Assembler uses more processing time (page 151).

Path length results are provided for over 60 REXX/VM coding suggestions to show the potential degree of benefit (page 155).

Measurement variability results are provided for the FS7F0R workload when run on a 9121-480 (page 180).

Changes That Affect Performance

This chapter contains descriptions of various changes to VM/ESA Release 2.1 that affect performance. Most of the changes are performance improvements and are listed under “Performance Improvements” on page 10. However, some have the potential to adversely affect performance. These are listed under “Performance Considerations” on page 14. The objectives of these two sections are as follows:

- Provide a comprehensive list of the significant performance changes.
- Allow installations to assess how their workloads may be affected by these changes.
- Describe new functions that applications could exploit to improve performance.

Throughout the rest of the report, various references are made to these changes. This further illustrates where these changes apply and how they may affect performance.

“Performance Management” on page 15 is the third section of this chapter. It discusses changes that affect VM/ESA performance management.

Performance Improvements

This section describes performance improvements made to CP, CMS, SFS, and VMSES/E in VM/ESA Release 2.1.

CP Improvements

The following are performance improvements made in CP.

Virtual Disk in Storage: VM/ESA Release 2.1 supports a new type of temporary minidisk known as a virtual disk in storage. A virtual disk in storage appears similar to a temporary disk (T-disk). However, virtual disks in storage can be shared. A virtual disk in storage is allocated in an ESA/370* address space in system storage instead of on DASD.

Virtual disks in storage enable guests, servers, and other applications to use standard FBA CCW programs and CP DIAGNOSE interfaces to exploit system storage for temporary data that traditionally resides on minidisks. Virtual reserve/release is supported for virtual disks in storage, which allows data sharing among multiple guests (such as VSE) that also support reserve/release CCWs.

Virtual disk in storage can provide better performance than traditional minidisks located on DASD. This improvement is mainly in terms of reduced elapsed time, although ITR also improves, which results from a reduction in real I/Os. The only real I/Os caused by the use of virtual disks in storage are in the form of page read and write operations that arise because the virtual disk in storage data contends for real storage and paging resources.

A small amount of frequently used data is an excellent candidate for using a virtual disk in storage. A VSE lock file, used to reflect locking status of shared resources for multiple VSE guests, is a good example. In this situation, the virtual disk in storage data causes a negligible increase in storage contention, the data becomes resident in real memory, and use of the data therefore causes a negligible amount of real I/O to occur.

For the VSE PACE workload, the external throughput rate (ETR) improved 1%-29% and the internal throughput rate (ITR) improved 2%-4%, depending on the particular use of virtual disks in storage and the system configuration.

Caution should be used when exploiting virtual disks in storage with large amounts of data. Overall system performance can still improve compared to the use of traditional minidisks, but only if the benefits that arise from avoiding real minidisk I/O outweigh the performance impact from increased storage use and paging. For more details, see "Virtual Disk in Storage" on page 76.

Load Wait State PSW: CP has been enhanced to reduce the path length associated with a guest virtual machine loading a wait state PSW.[1] This enhancement benefits servers, such as VTAM* and SFS, that do asynchronous processing, and guest operating systems that are run on non-dedicated processors. The benefits are in terms of

[1] This enhancement does not apply to the case where guests have dedicated processors, on a complex in basic mode, with the PR/SM* facility. In such cases this function is handled entirely under Start Interpretive Execution (SIE).

reduced processing requirements. The amount of benefit is proportional to the system-wide frequency of this operation, which is displayed as the GUESTWT field on the RTM SYSTEM screen. (This field was added to the SYSTEM screen as a modification to RTM Release 1.5.2; prior releases of RTM do not report this statistic.)

More than a 2% improvement in ITR has been observed for a V=R VSE guest (with dedicated I/O) running an I/O-intensive workload.

IUCV Processor Use: Processor use of the mainline functions (SEND/RECEIVE/REPLY) decreased by 7% in VM/ESA Release 2.1. Most of the improvement came from replacing calls to HCPIUG with an inline macro. The minidisk CMS-intensive and the SFS CMS-intensive workloads showed a 0.4% and a 0.5% ITR improvement, respectively.

Fast Path CCW Translation: Fast path CCW translation was extended to include support for the V=R full pack minidisk DASD environment. A VSE environment running the PACEX8 workload (heavy DASD I/O) showed a 25.4% improvement in ETR and a 21.0% improvement in ITR. See "9221-170 / V=R MODE=ESA with Full Pack Minidisks" on page 60 for more information.

In VM/ESA Release 2, support was available for the V=F and V=V full pack minidisk DASD environment. Refer to the *VM/ESA Release 2 Performance Report* for more information.

For background information on fast path CCW translation, see the *VM/ESA Release 1.1 Performance Report*, *VM/ESA Release 2 Performance Report*, and Washington System Center Flash 9220.

Shared Segment Serialization: Changes have been made to the CP algorithm that handles system data files containing saved segments and named saved systems (NSSs). The changes allow functions to run concurrently. Previously, they had to run consecutively. The following performance improvements are provided:

- Multiple saved segment loads, finds, and purges (using DIAGNOSE code X′64′ or functions that call it) can run at the same time. They can also run while a SAVESEG or SAVESYS command is being processed. Previously, only one saved segment or NSS operation could occur at any given time.
- Users can IPL NSSs and load saved segments into their virtual machines while other users are doing the same.
- Except in rare cases, saved segments and NSSs can be loaded, saved, or purged while the SPTAPE command is being used to dump or load a saved segment or NSS.
- A SAVESEG or SAVESYS operation can be interrupted with the PURGE NSS command. Previously, the purge had to wait for the save to complete.

Support for Data Compression: If VM/ESA is installed on an ES/9000 processor that supports hardware assisted data compression, any program that can use data compression when running natively can also use data compression when running as a guest of VM/ESA. The potential benefits of such data compression include reduced secondary storage requirements and reduced data transmission time.

CMS Improvements

The following section describes the performance improvements made in CMS.

EXECUTE XEDIT Macro: The EXECUTE XEDIT macro takes fewer instructions. The improvement was made by using some of the suggestions in "VM/ESA REXX Performance Guidelines" on page 155. FILELIST, DIRLIST, RDRLIST, MACLIST, CSLLIST, and CSLMAP are full screen commands that use the EXECUTE macro whenever a command is issued that applies to a line of information on the full screen display. The minidisk CMS-intensive workload showed a 0.5% improvement in ITR.

Minidisk File Cache Size: CMS uses a buffer called the "minidisk file cache" to perform read-ahead, write-behind processing for sequentially accessed files. (It is not the same as minidisk caching where CP uses expanded storage to cache 4KB-blocked CMS minidisks). The minidisk file cache reduces the number of minidisk file I/Os. When a file is read sequentially, CMS reads as many blocks at a time as will fit into the minidisk file cache. When a file is written sequentially, completed blocks are accumulated until they fill the file cache and are then written together.

In the past, the size of this cache was fixed at 8KB. The specified size only applies to minidisk I/O. There is (and has been) a separate file cache size value that applies to SFS files. That size defaults to 20KB and is also specified when the CMS nucleus is built.

8KB remains the default, but now a different size can be specified when the CMS nucleus is built. That size then applies to all users of that CMS nucleus. A minidisk file cache of up to 96KB can be specified. The actual size is, however, subject to a limit of 24 CMS data blocks. That means, for example, that the maximum file cache size is 48KB for a minidisk that is blocked at 2KB.

As the file cache size increases, minidisk I/Os decrease but system real storage requirements, paging, and DASD response time may increase. The 8KB default is suitable for storage-constrained systems, while a larger size is likely to result in an overall performance improvement for systems with more storage. See "Adjusting the Minidisk File Cache Size" on page 122 for more details.

The minidisk-only CMS-intensive workload showed almost a 1% improvement in ITR when increasing the file cache size from 8KB to 16KB in a storage constrained environment. However, there was no increase in paging.

CMS Multitasking Performance: Although not measured, it is believed that the performance of the CMS multitasking kernal will improve, reducing the processing requirements of CMS multitasking applications that make use of multiple virtual processors.

CMS Pipeline for Retrieving Accounting Data: The performance of the RETRIEVE utility for collecting CP accounting data is a recognized concern. The fact that RETRIEVE closes the output file after writing each accounting record is the biggest factor. APAR VM52043 was opened for this particular problem. The new STARSYS CMS Pipelines stage can be used to create a replacement for the RETRIEVE utility. In the STARSYS documentation, an example exec is shown using the *ACCOUNT system service. This example accepts a list of “precious” records. After writing a “precious” record, the exec closes the file. Otherwise, the file is closed when the internal buffers are filled. The performance of the example exec is similar to the RETRIEVE utility for “precious” records. However, it is significantly faster for the other records.

SFS Improvements

The following section describes performance improvements made in SFS.

SFS Control Data Backup: In the past, even if only one block of a 512-block cell had data in it, all of the blocks in that cell were backed up. SFS now backs up control data blocks that are allocated. This can reduce backup time and the amount of space required to hold the backup. It also means the amount of space required to hold the backup can now be estimated more accurately. The output from QUERY FILEPOOL MINIDISK (4K Blocks In-Use of the STORAGE GROUP MINIDISK INFORMATION) can be used to do this.

SFS Thread Blocking I/O: In VM/ESA Release 2, applications using the CMS multitasking interfaces could only do SFS requests asynchronously by using a polling and/or checking method to verify when the asynchronous request was complete. A thread can now issue an asynchronous SFS CSL request, do other processing, and then wait for completion of that SFS request by using the EventWait multitasking service. This means that it is now practical for CMS multitasking applications to use the SFS functions asynchronously.

VMSES/E Improvements

The following section describes performance improvements made in VMSES/E.

VMSES/E: The performance of the VMFBDNUC and VMFSIM functions has been improved.

The automation of more service processing in VMSES/E on VM/ESA Release 2.1 eliminates many manual tasks. Therefore, the overall time required to do these tasks will decrease. The following automation functions have been added to VMSES/E:

- Local Service Support

  The VMFSIM CHKLVL, VMFASM, VMFHASM, VMFHLASM, and VMFNLS commands provide a LOGMOD option to automatically update the Software Inventory when local service is applied.

- Automation of Library Generation

  The VMFBLD function automates the generation of callable services libraries and CMS/DOS libraries.

- Support for Generated Objects

  The VMFBLD function automates the building of generated objects, such as text decks.

Performance Considerations

This section describes changes in VM/ESA Release 2.1 that may adversely affect performance.

TRSAVE Enhancements: By use of the DEFERIO option on the TRSAVE command (privilege class A and C), the user can now request that trace buffers not be scheduled for I/O until tracing is stopped. This function will tend to be used with a large number of in-storage trace buffers in order to avoid wrapping the data. If the number of requested buffers is too large, use of this facility can adversely affect system performance.

3390 Model 9 DASD Support: A 3390 model 9 DASD rotates at one-third the speed of a 3390 model 3, resulting in a latency 3 times longer. Response time for the 3390 model 9 is 2 to 3 times that of the 3390 model 3. The 3390 model 9 is intended as a mass-storage device for applications which require faster access times than those provided by tape or optical drives, but which do not require the high performance of traditional DASD. These devices should not be used for system data or data used by applications that require high performance DASD.

FSWRITE Detect Filespace Limits: This new function prevents minidisk applications from receiving a SFS file space exceeded condition during file close processing. CMS was changed so that it checks a user′s SFS file space limit as files are being modified using the FSWRITE interface. If the application on the system heavily uses FSWRITE, this may adversely affect system performance.

This detection support caused a 0.05% decrease in internal throughput rate for the CMS-intensive SFS workload.

Performance Management

This following section discusses changes in VM/ESA Release 2.1 that affect VM/ESA performance management.

Effects on Accounting Data

The following list describes fields in the virtual machine resource usage accounting record (type 01) that may be affected by performance changes in VM/ESA Release 2.1. The columns where the field is located are shown in parentheses.

Milliseconds of processor time used (33-36) This is the total processor time charged to a user and includes both CP and emulation time. For most workloads, this should not change much as a result of the changes made in VM/ESA Release 2.1. Most CMS-intensive workloads are expected to experience little change in virtual processor time and a slight decrease in CP processor time.

Milliseconds of virtual processor time (37-40) This is the virtual time charged to a user. As mentioned above, little change is expected for most workloads. Virtual time no longer includes time to page in a frame from expanded storage. This may result in a decrease in this value (but not overall processor time) for storage-constrained systems that use expanded storage for paging.

Requested virtual nonspooled I/O starts (49-52) This is a total count of requests. All requests may not complete. The value of this field will tend to decrease if a minidisk file cache size larger than 8KB is selected. Virtual disk in storage I/Os are counted in this total.

Completed virtual nonspooled I/O starts (73-76) This is a total count of completed requests. The value of this field will tend to decrease if a minidisk file cache size larger than 8KB is selected. Virtual disk in storage I/Os are counted in this total.

Monitor Enhancements

The MONITOR LIST1403 file (on MAINT′s 194 minidisk) describes the content of each monitor record. Revision bars in the file show which records and fields are new in VM/ESA Release 2.1. Some of the more significant changes are summarized below. I/O assist monitor enhancements are reported in VMPRF 1.2.1 with APAR VM55672.

- Virtual Disks in Storage

  The monitor has been enhanced to include data on the virtual disk in storage function. Information is included for overall system usage, usage of each virtual disks in storage, and usage by individual user.

  - System limit values, number of virtual disks in storage, and current space allocated for virtual disks in storage are included in domain 0 record 7 (D0/R7).
  - A new record, "Virtual Disk in Storage Information," (D3/R17) was added. This sample record is created for each existing virtual disk in storage and includes address space name, owner, size, links, and I/Os.

- The count of virtual I/Os to a virtual disk in storage for a given user is included in user domain records D4/R2, D4/R3, and D4/R9. This value is a subset of the existing count of virtual DASD I/Os found in those same records.
- Using the address space name associated with the virtual disk in storage, the related storage domain records (D3/R12, D3/R13, D3/R14) can be used for additional information.

- I/O Assist Monitor Enhancements

  Prior to VM/ESA Release 2.1, the potential existed for anomalies in the monitor data associated with I/O assist in D6/R3. Each device eligible for I/O assist can be in one of three states: in assist, out of assist, or leaving assist. Counters for the transitions into each state and the time spent in each state are kept by CP. These counters are sampled and included in the D6/R3 monitor record. The anomalies resulted because the time-spent counter is not updated until a device leaves a state. Therefore, the monitor data did not include the time spent in the current state.

  The enhancement adds a flag indicating the current state of the device and a time stamp indicating when the device entered that state. Using this new information and the monitor record time stamp, one can compute the amount of time not included in time-spent counters and which state is missing this time.

- Improved Processor High Frequency Monitor Sampling

  High frequency state sampling is used for collecting some data found in D5/R3, such as the number of VMDBKs on Processor Local Dispatch Vector (PLDV). Prior to VM/ESA Release 2.1, this value was often artificially inflated for the master processor because, while collecting other high-frequency data, the monitor is running on the master and this sudden burst of activity skews the PLDV values. This has been corrected in VM/ESA Release 2.1 by sampling the processor data prior to the user data.

- Other Monitor Enhancements

  In this release, two event records (D3/R15 and D3/R16) have been added. These records are generated when a named saved system, discontiguous saved segment, or segment space is loaded into storage (accessed by the first user) or removed from storage (released by the last user). These records were added to fix anomalies with the existing sample record (D3/R3). The anomalies result from the counters used in the sample record being set back to zero when the last user releases the segment.

  This information can determine which segments should be preloaded. This prevents the shared segments′ page frames from becoming invalid when not in use to avoid page reads when the next user accesses the shared segments.

INDICATE Command

If INDICATE USER (privilege class E and G) is issued with the new EXPANDED option, the output is organized into groups (I/O, storage, processor, and so forth) and additional summary information is provided for the private and shared address spaces owned by that user. A number of the data fields have been enlarged to reduce data wrapping.

When INDICATE USER is issued without the EXPANDED option, a user′s counts for the I/O, RDR, PRT, and PCH fields are reset to zero whenever an ACNT command is processed for the user ID. With the EXPANDED option, these values are not reset. All reported time and count values are ever-increasing accumulators since logon.

INDICATE NSS (class E) is a new command that displays information about named saved systems and saved segments that are loaded in the system and are in use by at least one user. For each NSS or saved segment, the output includes paging statistics and the number of pages residing in main storage, expanded storage, and on DASD. INDICATE NSS makes available information that had been available on certain earlier VM releases by specifying INDICATE USER nss_name.

INDICATE SPACES (class E and G) is a new command that provides information about one or all of the address spaces owned by the specified user ID. For each address space, the output includes paging statistics and the number of pages residing in main storage, expanded storage, and on DASD.

QUERY VDISK Command

The QUERY VDISK command (privilege class B) displays a list of all existing virtual disks in storage. It can also display information about the system or user limits on the amount of virtual storage available for virtual disks in storage. Those limits are set using the SET VDISK command.

STARMONITOR

The STARMONITOR pipeline stage writes lines from the CP *MONITOR system service. STARMONITOR uses IUCV to connect to the CP *MONITOR system service. STARMONITOR writes these lines as logical records, beginning with the 20-byte prefix defined for monitor records. Monitor sample data, event data, or both can be chosen and records can be suppressed from one or more of these domains. A STARMONITOR stage can be used only as the first stage of a pipeline.

STARSYS

Use the STARSYS stage command to write lines from and send replies to a CP system service. STARSYS uses IUCV to connect to a two-way system service (such as *ACCOUNT, *LOGREC, or *SYMPTOM). A STARSYS stage can be used only as the first stage of a pipeline.

Measurement Information

This chapter discusses the types of processors used for measurements in the report, the level of software used, an explanation of the configuration details that are associated with each measurement, and the licensed programs and tools that were used in running and evaluating the performance measurements.

Hardware Used

The following processors and Licensed Internal Code levels were used for this report.

- 9121-742, Licensed Internal Code level C65051A

  This processor was used with 1, 2, 3, and 4 processors online and whatever processors were not being used were varied offline.

- 9121-480, Licensed Internal Code level C35674

  This processor was used for the 9121-480 and 9121-320 measurements. To run as a 9121-320, one processor was varied offline.

- 9221-170, Licensed Internal Code level C48069, and C49095

Software Used

Unless otherwise noted, a pre-general-availability level of VM/ESA Release 2.1 was used for the measurements in this report. All performance enhancements discussed in this report are part of the pre-general-availability level of code unless otherwise noted. Not all of the VM/ESA Release 2.1 measurements in this report were made with the same level of code. As the product developed, new code levels were made that supplanted the level that had been in use. In any evaluation section that compares VM/ESA Release 2.1 to itself the same level of code was maintained. Keep this in mind when trying to compare results that are taken from different sections.

Other releases of VM were used in the report. VM/ESA Release 2 was at the GA+first-RSU level (General Availability, Recommended Service Upgrade Tape). The service that was part of VM/ESA Release 2 after the first RSU level and forwarded to the pre-general-availability code can account for some of the difference between VM/ESA Release 2 and VM/ESA Release 2.1.

See the appropriate workload section in Appendix B, "Workloads" on page 191 for the other operating system and licensed programs′ software levels.

Format Description

This part of the report contains a general explanation of the configuration details that are associated with each measurement.

For each group of measurements there are five sections:

1. Workload: This specifies the name of the workload associated with the measurement. For more detail on the workload, see Appendix B, "Workloads" on page 191.

2. Hardware Configuration: This summarizes the hardware configuration, and it contains the following descriptions:

   - Processor model: The model of the processor.
   - Processors used: The number of processors used.
   - Storage: The amount of real and expanded storage used on the processor.
     – Real: The amount of real storage used on the processor.

       On the 9221, any real storage not defined for the specific measurement and not configured as expanded storage was attached to an idle user.

     – Expanded: The amount of expanded storage used on the processor.
   - Tape: The type of tape drive and the tape's purpose.
   - DASD: The DASD configuration used during the measurement.

     The table indicates the type of DASD actually used during the measurement, type of control units that connect these volumes to the system, the number of paths between the DASD and the control unit, and the distribution of these volumes for PAGE, SPOOL, TDSK, USER, SERVER and SYSTEM. R or W next to the DASD counts means Read or Write caching enabled respectively.

   - Communications: The type of control unit, number of communication control units, number of lines per control unit, and the line speed.

     The 3745-410 control unit used for this report was split in half and run in twin-dual mode, the number specified is the number of halves used for each of the processors (that is, the TPNS driver and the processor being measured). Each of these halves has a maximum of 50 lines available and can support a maximum of 3 000 users.

3. Software Configuration: This section contains pertinent software information.

   - Driver: The tool used to simulate users.
   - Think time distribution: The type of distribution used for the user think times.

     Bactrian This type of think time distribution represents a combination of both active and inactive user think times. The distribution includes long think times that occur when the user is not actively issuing commands. Actual user data were collected and used as input to the creation of the Bactrian distribution. This type of mechanism allows the transaction rate to vary depending on the command response times in the measurement.

     IOB This type of think time distribution represents the think time defined by the IBM Office Benchmark (IOB V2.1) workload. The think time includes an average 2-second delay between commands issued by TPNS, the built-in think times that are part of the IOB scripts, and the IOB script scheduling algorithm. The average message rate per user stays constant across all of the measurements. See "IBM Office Benchmark (IOB V2.1)" on page 203 for more details.

- CMS block size: The block size of the CMS minidisks.
- Virtual Machines: The virtual machines used in the measurement.

  For each virtual machine, the table indicates the following: name, number used, type, size and mode, share of the system resources scheduled, number of pages reserved, and any other options that were set.

4. Measurement Discussion: This contains an analysis of the performance data in the table and gives the overall performance findings.

5. Measurement Data: This contains the table of performance results. These data were obtained or derived from the tools listed in the section "Tools Description" on page 21 below.

   There are several cases where the same information is reported from two sources because the sources calculate the value in a slightly different manner. For example, consider the external throughput rate measures, ETR (T) and ETR, that are based on the command rate calculated by TPNS and RTM respectively. TPNS is outside the system and can directly count the command rate as it runs the commands in the scripts. Because RTM uses information that is inside the system, it has to make assumptions about when transactions begin and end. This can make the counts reported by RTM vary in meaning from run to run and vary from the values reported by TPNS. As a result, the analysis of the data relies more on the TPNS command rate. Furthermore, some values in the table (like TOT INT ADJ) were normalized to the TPNS command rate in an effort to get the most accurate performance measures possible.

   There are instances in these tables where two variables are equal yet there appears a non-zero number for their difference or percent difference. This indicates that the variables are only equal when they were rounded off to the significant digits that appear in the table.

   Performance terms listed in the tables and discussed in this part of the document are defined in the glossary.

Tools Description

A variety of licensed programs and internal tools were used to evaluate the performance measurements. The programs used in the measurements are listed below.

RTM Real Time Monitor records and reports performance data for VM systems. The appropriate level of RTM VM/ESA was used for VM/ESA Release 2 and VM/ESA Release 2.1 systems.

TPNS Teleprocessing Network Simulator is a terminal and network simulation tool.

VMPAF VM Performance Analysis Facility data are not in the measurement tables but the data are used for performance analysis of VM systems.

VMPRF VM Performance Reporting Facility is the VM monitor reduction program.

FSTTAPE Reduces hardware monitor data for the 9121.

Hardware Monitor Collects branch, event, and timing data.

MONFAST Collects branch, event, and timing data on a 9221 in addition to reducing the data it collects.

REDFP Consolidates the QUERY FILEPOOL STATUS data.

TPNS Reduction Program Reduces the TPNS log data to provide performance, load, and response time information.

Migration from VM/ESA Release 2

This chapter explores the performance effects of migrating from VM/ESA Release 2 to VM/ESA Release 2.1. The following environments were measured: CMS-intensive, OfficeVision, VSE guest, and VMSES/E.

DASD fast write was used for one OfficeVision environment, because OfficeVision′s workload had heavy write DASD I/O activity and a constraint on the amount of available DASD.

Each workload experiences some run variability. The degree of variability depends on the workload and configuration. A study of run variability was performed on the 9121-480 with the CMS-intensive minidisk version of the workload. The results are summarized in "Measurement Variability" on page 180.

CMS-Intensive

VM/ESA Release 2.1 had improved internal throughput rates and had improved or equivalent response times for the CMS-intensive environments measured. The ITR improvements resulting from decreased processor use can be attributed to the following functional enhancements:

- Load wait state PSW
- IUCV
- EXECUTE XEDIT macro.

For more information on these and other performance-related enhancements in VM/ESA Release 2.1, see "Changes That Affect Performance" on page 9.

The internal throughput rates and response times for these measurements are shown in the following 2 graphs.

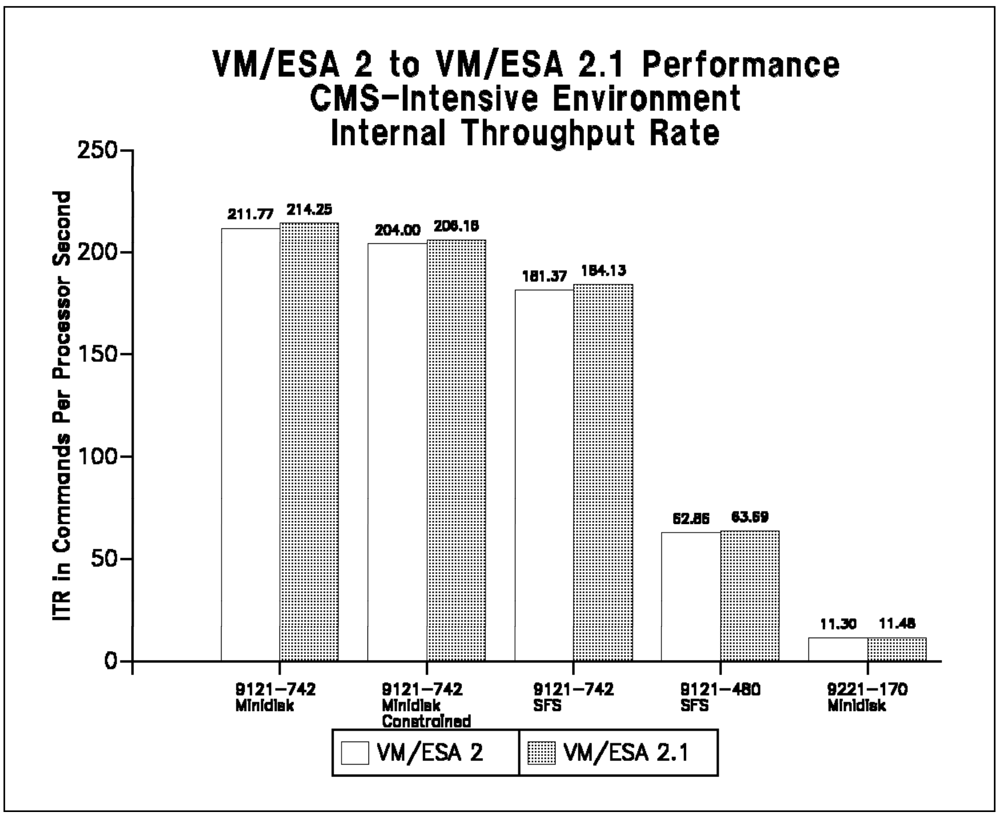

Figure 1. Internal throughput rate for the various CMS-intensive workloads. CMS-intensive environments compared between VM/ESA Release 2 and VM/ESA Release 2.1.

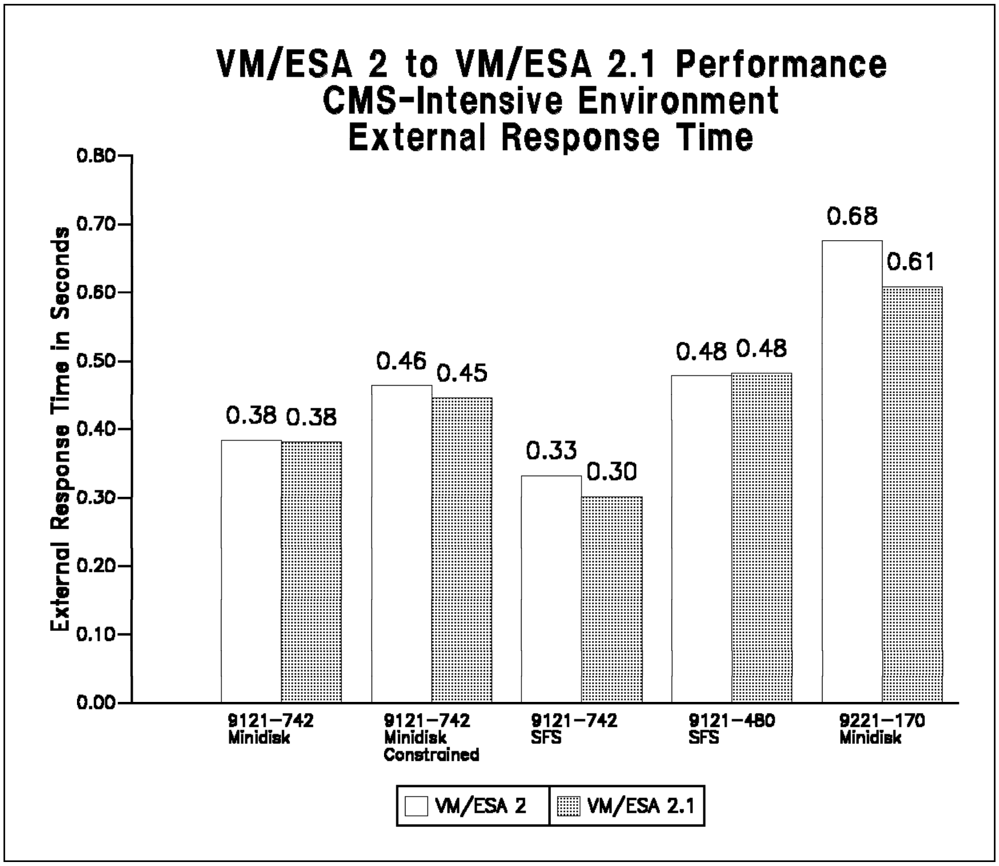

Figure 2. External response time for the various CMS-intensive workloads. CMS-intensive environments compared between VM/ESA Release 2 and VM/ESA Release 2.1.

9121-742 / Minidisk

Workload: FS7F0R

Hardware Configuration

Processor model:	9121-742
Processors used:	4
Storage	
Real:	1024MB
Expanded:	1024MB (64MB to 1024MB for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-2	2	6	5				
3390-2	3990-2	4	16	6	6			
3390-2	3990-3	4		2	7	32 R		2 R

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
SMART	1	RTM	16MB/370	3%	500	QUICKDSP ON
VSCSn	3	VSCS	64MB/XA	10000	1200	QUICKDSP ON
VTAMXA	1	VTAM	64MB/XA	10000	550	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	5400	User	3MB/XC	100		

Measurement Discussion: The following table shows that VM/ESA Release 2.1 compared to VM/ESA Release 2 remained equivalent in external response time (AVG LAST(T)), increased in internal throughput rate (ITR(H)), and decreased in paging (PAGE/CMD and XSTOR/CMD).

The internal throughput rate improved due to enhancements made in IUCV, load wait state PSW, and EXECUTE XEDIT macro. The first two improvements reduced VTAM's CP processor use (VTAM CP CPU/CMD(V)), which reduced overall CP processor use (CP/CMD (H)). The EXECUTE XEDIT macro improvement reduced virtual machine processor use (EMUL/CMD(H)).

The minidisk cache size was prevented from dropping below 64MB by issuing the command RETAIN XSTOR MDC 64M. The results from measurements discussed in “Minidisk Cache Tuning: Restricting the Arbiter” on page 146 show that the system performed better with the 64MB floor.

Table 1 (Page 1 of 3). Minidisk-only CMS-intensive migration from VM/ESA Release 2 on the 9121-742

Release **Run ID**	**VM/ESA 2** **S45E5400**	**VM/ESA 2.1** **S46E5400**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**5400**	**5400**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
Response Time				
TRIV INT	0.100	0.097	-0.003	-3.00%
NONTRIV INT	0.371	0.362	-0.009	-2.43%
TOT INT	0.261	0.254	-0.007	-2.68%
TOT INT ADJ	0.259	0.253	-0.006	-2.20%
AVG FIRST (T)	0.284	0.283	-0.002	-0.59%
AVG LAST (T)	0.384	0.381	-0.003	-0.78%
Throughput				
AVG THINK (T)	26.25	26.26	0.02	0.06%
ETR	187.48	188.50	1.02	0.54%
ETR (T)	189.22	189.31	0.08	0.04%
ETR RATIO	0.991	0.996	0.005	0.50%
ITR (H)	211.77	214.25	2.49	1.17%
ITR	52.47	53.40	0.93	1.78%
EMUL ITR	77.92	80.52	2.60	3.34%
ITRR (H)	1.000	1.012	0.012	1.17%
ITRR	1.000	1.018	0.018	1.78%
Proc. Usage				
PBT/CMD (H)	18.889	18.669	-0.219	-1.16%
PBT/CMD	18.867	18.647	-0.220	-1.16%
CP/CMD (H)	6.787	6.673	-0.114	-1.68%
CP/CMD	6.130	6.286	0.156	2.54%
EMUL/CMD (H)	12.102	11.996	-0.105	-0.87%
EMUL/CMD	12.736	12.361	-0.375	-2.95%
Processor Util.				
TOTAL (H)	357.42	353.42	-4.00	-1.12%
TOTAL	357.00	353.00	-4.00	-1.12%
UTIL/PROC (H)	89.36	88.36	-1.00	-1.12%
UTIL/PROC	89.25	88.25	-1.00	-1.12%
TOTAL EMUL (H)	229.00	227.10	-1.90	-0.83%
TOTAL EMUL	241.00	234.00	-7.00	-2.90%
MASTER TOTAL (H)	91.32	90.48	-0.85	-0.93%
MASTER TOTAL	91.00	90.00	-1.00	-1.10%
MASTER EMUL (H)	41.61	41.15	-0.46	-1.10%
MASTER EMUL	44.00	43.00	-1.00	-2.27%
TVR(H)	1.56	1.56	0.00	-0.29%
TVR	1.48	1.51	0.03	1.84%

Table 1 (Page 2 of 3). Minidisk-only CMS-intensive migration from VM/ESA Release 2 on the 9121-742				
Release **Run ID**	**VM/ESA 2** **S45E5400**	**VM/ESA 2.1** **S46E5400**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**5400**	**5400**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
Storage				
NUCLEUS SIZE (V)	2304KB	2368KB	64KB	2.78%
TRACE TABLE (V)	800KB	800KB	0KB	0.00%
WKSET (V)	70	69	-1	-1.43%
PGBLPGS	236 k	235 k	-1 k	-0.42%
PGBLPGS/USER	43.7	43.5	-0.2	-0.42%
FREEPGS	14315	14546	231	1.61%
FREE UTIL	0.91	0.92	0.00	0.34%
SHRPGS	1453	1493	40	2.75%
Paging				
READS/SEC	542	542	0	0.00%
WRITES/SEC	339	325	-14	-4.13%
PAGE/CMD	4.656	4.580	-0.076	-1.63%
PAGE IO RATE (V)	122.100	117.100	-5.000	-4.10%
PAGE IO/CMD (V)	0.645	0.619	-0.027	-4.14%
XSTOR IN/SEC	891	888	-3	-0.34%
XSTOR OUT/SEC	1345	1326	-19	-1.41%
XSTOR/CMD	11.817	11.695	-0.121	-1.03%
FAST CLR/CMD	6.516	6.529	0.013	0.20%
Queues				
DISPATCH LIST	104.18	101.98	-2.20	-2.11%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	1652	1653	1	0.06%
VIO/CMD	8.730	8.732	0.001	0.02%
RIO RATE (V)	575	576	1	0.17%
RIO/CMD (V)	3.039	3.043	0.004	0.13%
MDC READS	1048	1047	-1	-0.10%
MDC WRITES	519	518	-1	-0.19%
MDC MODS	421	421	0	0.00%
MDC HIT RATIO	0.91	0.91	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	20.408	20.529	0.121	0.59%
DIAG/CMD	24.894	24.845	-0.049	-0.20%
DIAG 08/CMD	0.779	0.760	-0.019	-2.43%
DIAG 14/CMD	0.025	0.025	0.000	0.13%
DIAG 58/CMD	1.251	1.250	-0.001	-0.05%
DIAG 98/CMD	0.347	0.363	0.016	4.61%
DIAG A4/CMD	3.907	3.901	-0.006	-0.15%
DIAG A8/CMD	1.879	1.889	0.010	0.53%
DIAG 214/CMD	13.014	12.964	-0.049	-0.38%
SIE/CMD	52.472	52.539	0.067	0.13%
SIE INTCPT/CMD	35.681	35.727	0.046	0.13%
FREE TOTL/CMD	63.417	63.389	-0.028	-0.04%

Table 1 (Page 3 of 3). Minidisk-only CMS-intensive migration from VM/ESA Release 2 on the 9121-742				
Release **Run ID**	**VM/ESA 2** **S45E5400**	**VM/ESA 2.1** **S46E5400**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**5400**	**5400**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
VTAM Machines				
WKSET (V)	4096	4108	12	0.29%
TOT CPU/CMD (V)	2.7628	2.6471	-0.1157	-4.19%
CP CPU/CMD (V)	1.3770	1.2737	-0.1033	-7.50%
VIRT CPU/CMD (V)	1.3858	1.3734	-0.0124	-0.89%
DIAG 98/CMD (V)	0.347	0.363	0.016	4.62%
Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM				

9121-742 / Minidisk / Storage-Constrained

Workload: FS7F0R

Hardware Configuration

Processor model:	9121-742
Processors used:	4
Storage	
Real:	320MB
Expanded:	64MB (all for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-2	2	6	5				
3390-2	3990-2	4	16	6	6			
3390-2	3990-3	4		2	7	32 R		2 R

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
SMART	1	RTM	16MB/370	3%	500	QUICKDSP ON
VSCSn	3	VSCS	64MB/XA	10000	1200	QUICKDSP ON
VTAMXA	1	VTAM	64MB/XA	10000	550	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	5200	User	3MB/XC	100		

Measurement Discussion: The following table shows that VM/ESA Release 2.1 compared to VM/ESA Release 2 remained equivalent in response time (AVG LAST(T)), increased in internal throughput rate (ITR(H)), and decreased in paging (PAGE/CMD).

The internal throughput rate results are similar to those reported in "9121-742 / Minidisk" on page 25. Refer to that section for details.

Table 2 (Page 1 of 2). Minidisk-only storage-constrained CMS-intensive migration from VM/ESA Release 2 on the 9121-742				
Release **Run ID**	**VM/ESA 2** **S45E5201**	**VM/ESA 2.1** **S46E5200**	**Difference**	**% Difference**
Environment				
Real Storage	**320MB**	**320MB**		
Exp. Storage	**64MB**	**64MB**		
Users	**5200**	**5200**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
Response Time				
TRIV INT	0.178	0.178	0.000	0.00%
NONTRIV INT	0.533	0.526	-0.007	-1.31%
TOT INT	0.403	0.401	-0.002	-0.50%
TOT INT ADJ	0.372	0.366	-0.007	-1.86%
AVG FIRST (T)	0.339	0.321	-0.018	-5.31%
AVG LAST (T)	0.464	0.446	-0.018	-3.88%
Throughput				
AVG THINK (T)	26.28	26.29	0.01	0.03%
ETR	167.92	165.86	-2.06	-1.23%
ETR (T)	181.68	181.95	0.27	0.15%
ETR RATIO	0.924	0.912	-0.013	-1.37%
ITR (H)	204.00	206.16	2.16	1.06%
ITR	47.16	47.05	-0.12	-0.25%
EMUL ITR	72.36	72.34	-0.02	-0.02%
ITRR (H)	1.000	1.011	0.011	1.06%
ITRR	1.000	0.998	-0.002	-0.25%
Proc. Usage				
PBT/CMD (H)	19.608	19.403	-0.205	-1.05%
PBT/CMD	19.595	19.401	-0.194	-0.99%
CP/CMD (H)	7.340	7.196	-0.144	-1.96%
CP/CMD	6.825	6.815	-0.010	-0.15%
EMUL/CMD (H)	12.268	12.206	-0.062	-0.50%
EMUL/CMD	12.770	12.586	-0.184	-1.44%
Processor Util.				
TOTAL (H)	356.24	353.02	-3.22	-0.90%
TOTAL	356.00	353.00	-3.00	-0.84%
UTIL/PROC (H)	89.06	88.26	-0.80	-0.90%
UTIL/PROC	89.00	88.25	-0.75	-0.84%
TOTAL EMUL (H)	222.89	222.09	-0.80	-0.36%
TOTAL EMUL	232.00	229.00	-3.00	-1.29%
MASTER TOTAL (H)	90.70	89.89	-0.81	-0.90%
MASTER TOTAL	91.00	90.00	-1.00	-1.10%
MASTER EMUL (H)	39.96	39.54	-0.42	-1.05%
MASTER EMUL	42.00	41.00	-1.00	-2.38%
TVR(H)	1.60	1.59	-0.01	-0.55%
TVR	1.53	1.54	0.01	0.46%
Storage				
NUCLEUS SIZE (V)	2304KB	2368KB	64KB	2.78%
TRACE TABLE (V)	800KB	800KB	0KB	0.00%
WKSET (V)	73	72	-1	-1.37%
PGBLPGS	57676	57344	-332	-0.58%
PGBLPGS/USER	11.1	11.0	-0.1	-0.58%
FREEPGS	13824	14122	298	2.16%
FREE UTIL	0.91	0.92	0.02	1.89%
SHRPGS	1451	1462	11	0.76%

Table 2 (Page 2 of 2). Minidisk-only storage-constrained CMS-intensive migration from VM/ESA Release 2 on the 9121-742

Release **Run ID**	**VM/ESA 2** **S45E5201**	**VM/ESA 2.1** **S46E5200**	**Difference**	**% Difference**
Environment				
Real Storage	**320MB**	**320MB**		
Exp. Storage	**64MB**	**64MB**		
Users	**5200**	**5200**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
Paging				
READS/SEC	1958	1904	-54	-2.76%
WRITES/SEC	1467	1450	-17	-1.16%
PAGE/CMD	18.852	18.434	-0.418	-2.22%
PAGE IO RATE (V)	650.100	634.800	-15.300	-2.35%
PAGE IO/CMD (V)	3.578	3.489	-0.089	-2.50%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.489	6.491	0.001	0.02%
Queues				
DISPATCH LIST	121.16	122.20	1.04	0.86%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	1591	1594	3	0.19%
VIO/CMD	8.757	8.761	0.004	0.04%
RIO RATE (V)	1090	1076	-14	-1.28%
RIO/CMD (V)	6.000	5.914	-0.086	-1.43%
MDC READS	1008	1007	-1	-0.10%
MDC WRITES	499	498	-1	-0.20%
MDC MODS	406	405	-1	-0.25%
MDC HIT RATIO	0.91	0.91	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	20.952	21.161	0.209	1.00%
DIAG/CMD	24.967	24.997	0.031	0.12%
DIAG 08/CMD	0.780	0.768	-0.012	-1.49%
DIAG 14/CMD	0.025	0.025	0.000	-0.02%
DIAG 58/CMD	1.251	1.251	0.000	0.03%
DIAG 98/CMD	0.366	0.385	0.019	5.18%
DIAG A4/CMD	3.916	3.908	-0.008	-0.20%
DIAG A8/CMD	1.885	1.896	0.011	0.56%
DIAG 214/CMD	13.003	13.008	0.004	0.03%
SIE/CMD	53.165	53.225	0.060	0.11%
SIE INTCPT/CMD	36.684	37.257	0.573	1.56%
FREE TOTL/CMD	60.546	65.954	5.408	8.93%
VTAM Machines				
WKSET (V)	4003	3994	-9	-0.22%
TOT CPU/CMD (V)	2.9172	2.8030	-0.1142	-3.91%
CP CPU/CMD (V)	1.4586	1.3435	-0.1151	-7.89%
VIRT CPU/CMD (V)	1.4586	1.4595	0.0009	0.06%
DIAG 98/CMD (V)	0.366	0.385	0.019	5.15%

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

9121-742 / SFS

Workload: FS7FMAXR

Hardware Configuration

Processor model:	9121-742
Processors used:	4
Storage	
Real:	1024MB
Expanded:	1024MB (64MB to 1024MB for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-2	2	6	1				
3390-2	3990-2	4	16	6	6			
3390-2	3990-3	4		2	7		32 R	2 R

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
CRRSERV1	1	SFS	16MB/XC	100		
ROSERV1	1	SFS	32MB/XC	100		QUICKDSP ON
RWSERVn	4	SFS	64MB/XC	1500	2100	QUICKDSP ON
SMART	1	RTM	16MB/370	3%	600	QUICKDSP ON
VSCSn	3	VSCS	64MB/XA	10000	1100	QUICKDSP ON
VTAMXA	1	VTAM	64MB/XA	10000	600	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	4560	User	3MB/XC	100		

Measurement Discussion: The FS7FMAXR workload has all user data in SFS files. The read/write files are in FILECONTROL directories, and the read-only files are in DIRCONTROL directories that are in VM Data Spaces. The read-only files reside in a separate read-only file pool. The system files are on S and Y read-only minidisks.

The performance of VM/ESA Release 2.1 showed improvement over that of VM/ESA Release 2. The SFS workload showed lower response time, increased throughput rate, and equivalent paging.

There was an improvement in both internal (TOT INT ADJ) and external (AVG LAST(T)) response times of 3.6% and 9.0% respectively. The response time improvement was attributed to the IUCV, load wait state PSW simulation, and EXECUTE XEDIT macro performance enhancements.

There was an increase in internal throughput rate (ITR (H)) from VM/ESA Release 2 to VM/ESA Release 2.1. As with the response time improvement, this improvement was attributed to the IUCV, load wait state PSW simulation, and EXECUTE XEDIT macro performance enhancements. As shown in the following table, this increase was 1.5% for the SFS environment. The internal throughput rate improved more in this environment compared to the mindisk-only environment (see "9121-742 / Minidisk" on page 25) because the SFS server activity increases the frequency of IUCV/APPC usage and load wait state PSW simulation. This was also reflected by the 6.9% decrease in CP CPU/CMD (V) for the SFS servers.

Table 3 (Page 1 of 3). SFS CMS-intensive migration from VM/ESA Release 2 on the 9121-742

Release **Run ID**	**VM/ESA 2** **S45S4562**	**VM/ESA 2.1** **S46S4561**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**4560**	**4560**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
Response Time				
TRIV INT	0.098	0.099	0.001	1.02%
NONTRIV INT	0.349	0.339	-0.010	-2.87%
TOT INT	0.254	0.252	-0.002	-0.79%
TOT INT ADJ	0.240	0.232	-0.009	-3.62%
AVG FIRST (T)	0.243	0.218	-0.025	-10.30%
AVG LAST (T)	0.332	0.302	-0.030	-9.03%
Throughput				
AVG THINK (T)	26.23	26.21	-0.02	-0.06%
ETR	151.22	147.14	-4.08	-2.70%
ETR (T)	159.89	160.15	0.27	0.17%
ETR RATIO	0.946	0.919	-0.027	-2.86%
ITR (H)	181.37	184.13	2.76	1.52%
ITR	42.92	42.32	-0.60	-1.39%
EMUL ITR	64.39	64.02	-0.37	-0.57%
ITRR (H)	1.000	1.015	0.015	1.52%
ITRR	1.000	0.986	-0.014	-1.39%

Table 3 (Page 2 of 3). SFS CMS-intensive migration from VM/ESA Release 2 on the 9121-742				
Release **Run ID**	**VM/ESA 2** **S45S4562**	**VM/ESA 2.1** **S46S4561**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**4560**	**4560**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
Proc. Usage				
PBT/CMD (H)	22.054	21.724	-0.330	-1.50%
PBT/CMD	22.078	21.730	-0.349	-1.58%
CP/CMD (H)	8.005	7.778	-0.227	-2.83%
CP/CMD	7.380	7.368	-0.012	-0.17%
EMUL/CMD (H)	14.050	13.946	-0.103	-0.74%
EMUL/CMD	14.698	14.361	-0.337	-2.29%
Processor Util.				
TOTAL (H)	352.62	347.91	-4.70	-1.33%
TOTAL	353.00	348.00	-5.00	-1.42%
UTIL/PROC (H)	88.15	86.98	-1.18	-1.33%
UTIL/PROC	88.25	87.00	-1.25	-1.42%
TOTAL EMUL (H)	224.63	223.35	-1.28	-0.57%
TOTAL EMUL	235.00	230.00	-5.00	-2.13%
MASTER TOTAL (H)	89.78	88.73	-1.05	-1.17%
MASTER TOTAL	90.00	89.00	-1.00	-1.11%
MASTER EMUL (H)	43.98	43.28	-0.70	-1.59%
MASTER EMUL	46.00	45.00	-1.00	-2.17%
TVR(H)	1.57	1.56	-0.01	-0.77%
TVR	1.50	1.51	0.01	0.73%
Storage				
NUCLEUS SIZE (V)	2304KB	2368KB	64KB	2.78%
TRACE TABLE (V)	800KB	800KB	0KB	0.00%
WKSET (V)	67	67	0	0.00%
PGBLPGS	239 k	239 k	0 k	0.00%
PGBLPGS/USER	52.4	52.4	0.0	0.00%
FREEPGS	12323	12606	283	2.30%
FREE UTIL	0.91	0.91	0.00	-0.02%
SHRPGS	1673	1637	-36	-2.15%
Paging				
READS/SEC	371	370	-1	-0.27%
WRITES/SEC	186	183	-3	-1.61%
PAGE/CMD	3.484	3.453	-0.031	-0.88%
PAGE IO RATE (V)	53.100	53.000	-0.100	-0.19%
PAGE IO/CMD (V)	0.332	0.331	-0.001	-0.35%
XSTOR IN/SEC	761	772	11	1.45%
XSTOR OUT/SEC	1033	1031	-2	-0.19%
XSTOR/CMD	11.221	11.258	0.038	0.34%
FAST CLR/CMD	6.724	6.706	-0.017	-0.26%
Queues				
DISPATCH LIST	87.53	83.34	-4.19	-4.78%
ELIGIBLE LIST	0.00	0.00	0.00	na

Table 3 (Page 3 of 3). SFS CMS-intensive migration from VM/ESA Release 2 on the 9121-742

Release Run ID	VM/ESA 2 S45S4562	VM/ESA 2.1 S46S4561	Difference	% Difference
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**4560**	**4560**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
I/O				
VIO RATE	1373	1374	1	0.07%
VIO/CMD	8.587	8.579	-0.008	-0.09%
RIO RATE (V)	485	477	-8	-1.65%
RIO/CMD (V)	3.033	2.978	-0.055	-1.81%
MDC READS	792	806	14	1.77%
MDC WRITES	331	314	-17	-5.14%
MDC MODS	227	225	-2	-0.88%
MDC HIT RATIO	0.88	0.90	0.02	2.27%
PRIVOPs				
PRIVOP/CMD	28.297	28.293	-0.004	-0.01%
DIAG/CMD	22.375	22.265	-0.110	-0.49%
DIAG 08/CMD	0.780	0.759	-0.021	-2.68%
DIAG 14/CMD	0.025	0.025	0.000	-0.24%
DIAG 58/CMD	1.250	1.250	0.000	0.02%
DIAG 98/CMD	0.476	0.507	0.031	6.43%
DIAG A4/CMD	2.121	2.100	-0.021	-0.99%
DIAG A8/CMD	1.633	1.638	0.005	0.30%
DIAG 214/CMD	12.405	12.280	-0.125	-1.00%
SIE/CMD	60.218	60.300	0.081	0.14%
SIE INTCPT/CMD	43.357	43.416	0.059	0.14%
FREE TOTL/CMD	68.799	68.685	-0.114	-0.17%
VTAM Machines				
WKSET (V)	3720	3714	-6	-0.16%
TOT CPU/CMD (V)	2.9171	2.8064	-0.1107	-3.79%
CP CPU/CMD (V)	1.4481	1.3390	-0.1091	-7.53%
VIRT CPU/CMD (V)	1.4690	1.4674	-0.0016	-0.11%
DIAG 98/CMD (V)	0.476	0.506	0.030	6.35%
SFS Servers				
WKSET (V)	9542	9541	-1	-0.01%
TOT CPU/CMD (V)	2.5317	2.4491	-0.0826	-3.26%
CP CPU/CMD (V)	1.1773	1.0962	-0.0811	-6.89%
VIRT CPU/CMD (V)	1.3544	1.3529	-0.0015	-0.11%
FP REQ/CMD(Q)	1.279	1.274	-0.005	-0.39%
IO/CMD (Q)	1.803	1.733	-0.070	-3.88%
IO TIME/CMD (Q)	0.023	0.022	-0.001	-4.35%
SFS TIME/CMD (Q)	0.031	0.031	0.000	0.00%

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Q=Query Filepool Counters, Unmarked=RTM

9121-480 / SFS

Workload: FS7FMAXR

Hardware Configuration

Processor model:	9121-480
Processors used:	2
Storage	
Real:	224MB
Expanded:	32MB (all for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-2	4	16	4	6			
3390-2	3990-3	2						2 R
3390-2	3990-3	4		1	2		16 R	

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3745-410	1	42	56Kb

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
CRRSERV1	1	SFS	16MB/XC	100		
ROSERV1	1	SFS	32MB/XC	100		QUICKDSP ON
RWSERVn	2	SFS	64MB/XC	1500	1300	QUICKDSP ON
SMART	1	RTM	16MB/370	3%	300	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	600	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	1620	User	3MB/XC	100		

Measurement Discussion: For these measurements 3745-410s were used for communications.

The results were similiar to those observed on the 9121-742 (refer to "9121-742 / SFS" on page 32) except the external response time did not decrease for VM/ESA Release 2.1 compared to VM/ESA Release 2.

There was an improvement in internal response time (TOT INT ADJ) of 2.5%. The response time improvement was attributed to the IUCV, load wait state PSW simulation, and EXECUTE XEDIT macro performance enhancements. The external response time (AVG LAST (T)) was essentially unchanged. In this environment, the beneficial effects of the performance enhancements was offset by an increase in paging in the VTAM machine.

There was a 1.3% increase in internal throughput rate (ITR (H)) from VM/ESA Release 2 to VM/ESA Release 2.1. As with the internal response time improvement, this was attributed to the IUCV, load wait state PSW simulation, and EXECUTE XEDIT macro performance enhancements. The increase in internal throughput rate is slightly less than observed on the 9121-742 because on the 9121-480 there are no external VSCS machines so IUCV traffic is reduced.

Table 4 (Page 1 of 3). SFS CMS-intensive migration from VM/ESA Release 2 on the 9121-480

Release **Run ID**	**VM/ESA 2** **L25S1620**	**VM/ESA 2.1** **L26S1620**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1620**	**1620**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.052	0.051	-0.001	-1.92%
NONTRIV INT	0.455	0.444	-0.011	-2.42%
TOT INT	0.253	0.246	-0.007	-2.77%
TOT INT ADJ	0.320	0.312	-0.008	-2.49%
AVG FIRST (T)	0.267	0.274	0.007	2.62%
AVG LAST (T)	0.478	0.482	0.004	0.84%
Throughput				
AVG THINK (T)	26.54	26.49	-0.05	-0.19%
ETR	71.07	70.93	-0.14	-0.20%
ETR (T)	56.19	55.92	-0.27	-0.48%
ETR RATIO	1.265	1.268	0.004	0.29%
ITR (H)	62.86	63.69	0.83	1.32%
ITR	39.76	40.41	0.66	1.65%
EMUL ITR	59.51	60.32	0.81	1.36%
ITRR (H)	1.000	1.013	0.013	1.32%
ITRR	1.000	1.017	0.017	1.65%
Proc. Usage				
PBT/CMD (H)	31.818	31.403	-0.415	-1.31%
PBT/CMD	31.857	31.474	-0.382	-1.20%
CP/CMD (H)	11.297	10.972	-0.326	-2.88%
CP/CMD	10.678	10.372	-0.306	-2.87%
EMUL/CMD (H)	20.521	20.431	-0.090	-0.44%
EMUL/CMD	21.178	21.102	-0.076	-0.36%

Table 4 (Page 2 of 3). SFS CMS-intensive migration from VM/ESA Release 2 on the 9121-480

Release **Run ID**	**VM/ESA 2** **L25S1620**	**VM/ESA 2.1** **L26S1620**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1620**	**1620**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Processor Util.				
TOTAL (H)	178.79	175.60	-3.18	-1.78%
TOTAL	179.00	176.00	-3.00	-1.68%
UTIL/PROC (H)	89.39	87.80	-1.59	-1.78%
UTIL/PROC	89.50	88.00	-1.50	-1.68%
TOTAL EMUL (H)	115.31	114.25	-1.06	-0.92%
TOTAL EMUL	119.00	118.00	-1.00	-0.84%
MASTER TOTAL (H)	89.32	87.85	-1.47	-1.65%
MASTER TOTAL	89.00	88.00	-1.00	-1.12%
MASTER EMUL (H)	52.92	52.32	-0.60	-1.14%
MASTER EMUL	55.00	54.00	-1.00	-1.82%
TVR(H)	1.55	1.54	-0.01	-0.87%
TVR	1.50	1.49	-0.01	-0.84%
Storage				
NUCLEUS SIZE (V)	2300KB	2364KB	64KB	2.78%
TRACE TABLE (V)	400KB	400KB	0KB	0.00%
WKSET (V)	79	79	0	0.00%
PGBLPGS	48674	48537	-137	-0.28%
PGBLPGS/USER	30.0	30.0	-0.1	-0.28%
FREEPGS	4389	4495	106	2.42%
FREE UTIL	0.90	0.90	0.00	0.22%
SHRPGS	1245	1239	-6	-0.48%
Paging				
READS/SEC	522	522	0	0.00%
WRITES/SEC	348	347	-1	-0.29%
PAGE/CMD	15.483	15.540	0.057	0.37%
PAGE IO RATE (V)	132.500	127.200	-5.300	-4.00%
PAGE IO/CMD (V)	2.358	2.275	-0.083	-3.54%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.727	6.778	0.050	0.75%
Queues				
DISPATCH LIST	39.77	39.51	-0.26	-0.66%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	486	484	-2	-0.41%
VIO/CMD	8.649	8.655	0.006	0.07%
RIO RATE (V)	302	297	-5	-1.66%
RIO/CMD (V)	5.375	5.311	-0.063	-1.18%
MDC READS	271	268	-3	-1.11%
MDC WRITES	128	129	1	0.78%
MDC MODS	80	80	0	0.00%
MDC HIT RATIO	0.84	0.84	0.00	0.00%

Table 4 (Page 3 of 3). SFS CMS-intensive migration from VM/ESA Release 2 on the 9121-480				
Release **Run ID**	**VM/ESA 2** **L25S1620**	**VM/ESA 2.1** **L26S1620**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1620**	**1620**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
PRIVOPs				
PRIVOP/CMD	25.048	24.847	-0.201	-0.80%
DIAG/CMD	23.526	23.371	-0.155	-0.66%
DIAG 08/CMD	0.780	0.760	-0.020	-2.58%
DIAG 14/CMD	0.025	0.025	0.000	-0.19%
DIAG 58/CMD	1.251	1.250	-0.001	-0.05%
DIAG 98/CMD	0.413	0.418	0.005	1.15%
DIAG A4/CMD	2.123	2.101	-0.023	-1.06%
DIAG A8/CMD	1.628	1.648	0.020	1.24%
DIAG 214/CMD	12.366	12.213	-0.153	-1.23%
SIE/CMD	58.143	58.031	-0.112	-0.19%
SIE INTCPT/CMD	41.281	41.202	-0.080	-0.19%
FREE TOTL/CMD	70.921	71.515	0.594	0.84%
VTAM Machines				
WKSET (V)	617	631	14	2.27%
TOT CPU/CMD (V)	3.6682	3.5269	-0.1413	-3.85%
CP CPU/CMD (V)	1.6215	1.4903	-0.1312	-8.09%
VIRT CPU/CMD (V)	2.0467	2.0367	-0.0100	-0.49%
DIAG 98/CMD (V)	0.413	0.418	0.005	1.21%
SFS Servers				
WKSET (V)	3201	3204	3	0.09%
TOT CPU/CMD (V)	3.8263	3.7256	-0.1007	-2.63%
CP CPU/CMD (V)	1.8786	1.7486	-0.1300	-6.92%
VIRT CPU/CMD (V)	1.9478	1.9771	0.0293	1.50%
FP REQ/CMD(Q)	1.275	1.269	-0.006	-0.47%
IO/CMD (Q)	1.931	1.891	-0.040	-2.07%
IO TIME/CMD (Q)	0.024	0.024	0.000	0.00%
SFS TIME/CMD (Q)	0.032	0.032	0.000	0.00%
Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Q=Query Filepool Counters, Unmarked=RTM				

9221-170 / Minidisk

Workload: FS7F0R

Hardware Configuration

Processor model: 9221-170
Processors used: 1
Storage
  Real: 48MB
  Expanded: 16MB (all for MDC)
Tape: 3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-2	1	16	5	6			
3390-2	3990-3	1		2	2	8 R		2 R

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver: TPNS
Think time distribution: Bactrian
CMS block size: 4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
MAINT	1	support	30MB/XA	100		
SMART	1	RTM	16MB/370	3%	350	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	300	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	290	User	3MB/XC	100		

Measurement Discussion: The following table shows that VM/ESA Release 2.1 compared to VM/ESA Release 2 improved in response time (AVG LAST(T)), increased in internal throughput rate (ITR(H)), and decreased in paging (PAGE/CMD).

The measurements followed the 9221 tuning recommended in the *VM/ESA Release 1.1 Performance Report*. This tuning concentrates on reducing the number of I/Os and SIE instructions to significantly improve internal throughput rate and response time for the 9221. In particular

- configure a portion of real storage as expanded storage and use it exclusively for minidisk cache. The result is a replacement of DASD I/Os with less processor-intensive minidisk cache reads.
- set the DELAY operand in the VTAM CTCA channel-attachment major node to 0.2. This reduces VTAM I/Os and the associated SIE interceptions.
- set IPOLL ON for VTAM. This reduces the number of IUCV instructions and the associated SIE interceptions.
- preload shared segments (FORTRAN and DCF) from an idle user. This prevents the shared segments′ page frames from becoming invalid when not in use to avoid page reads when the next user accesses the shared segments.

Overall CP processor usage (CP/CMD (H)) did not improve as much on the 9221-170 as on the 9121-742. The 9221-170 has one internal VSCS as opposed to three external VSCS machines for the 9121-742. Therefore, IUCV traffic is reduced and improvements made in IUCV and load wait state PSW are not as pronounced.

The improvement in response time was greater in this environment than for the 9121-742. A smaller processor, such as the 9221-170, shows a greater response time improvement when processor use (PBT/CMD(H)) decreased, and waiting in queue to use the processor is reduced. On smaller systems, they comprise a bigger percentage of overall response time.

Table 5 (Page 1 of 2). Minidisk-only CMS-intensive migration from VM/ESA Release 2 on the 9221-170

Release **Run ID**	**VM/ESA 2** **H15E0290**	**VM/ESA 2.1** **H16E0290**	**Difference**	**% Difference**
Environment				
Real Storage	**48MB**	**48MB**		
Exp. Storage	**16MB**	**16MB**		
Users	**290**	**290**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**1**	**1**		
Response Time				
TRIV INT	0.157	0.154	-0.003	-1.91%
NONTRIV INT	0.965	0.861	-0.104	-10.78%
TOT INT	0.729	0.655	-0.074	-10.15%
TOT INT ADJ	0.623	0.559	-0.064	-10.33%
AVG FIRST (T)	0.297	0.281	-0.016	-5.39%
AVG LAST (T)	0.676	0.609	-0.067	-9.91%
Throughput				
AVG THINK (T)	28.17	28.20	0.03	0.11%
ETR	8.63	8.62	-0.01	-0.12%
ETR (T)	10.09	10.10	0.01	0.08%
ETR RATIO	0.855	0.853	-0.002	-0.20%
ITR (H)	11.30	11.48	0.18	1.61%
ITR	9.66	9.80	0.14	1.47%
EMUL ITR	14.66	14.99	0.33	2.25%
ITRR (H)	1.000	1.016	0.016	1.61%
ITRR	1.000	1.015	0.015	1.47%
Proc. Usage				
PBT/CMD (H)	88.515	87.113	-1.401	-1.58%
PBT/CMD	88.196	87.131	-1.065	-1.21%
CP/CMD (H)	36.183	35.687	-0.497	-1.37%
CP/CMD	29.729	29.704	-0.025	-0.08%
EMUL/CMD (H)	52.331	51.427	-0.905	-1.73%
EMUL/CMD	58.467	57.427	-1.040	-1.78%
Processor Util.				
TOTAL (H)	89.32	87.98	-1.34	-1.50%
TOTAL	89.00	88.00	-1.00	-1.12%
TOTAL EMUL (H)	52.81	51.94	-0.87	-1.65%
TOTAL EMUL	59.00	58.00	-1.00	-1.69%
TVR(H)	1.69	1.69	0.00	0.15%
TVR	1.51	1.52	0.01	0.58%
Storage				
NUCLEUS SIZE (V)	2304KB	2368KB	64KB	2.78%
TRACE TABLE (V)	200KB	200KB	0KB	0.00%
WKSET (V)	88	84	-4	-4.55%
PGBLPGS	9598	9562	-36	-0.38%
PGBLPGS/USER	33.1	33.0	-0.1	-0.38%
FREEPGS	848	861	13	1.53%
FREE UTIL	0.89	0.90	0.01	0.80%
SHRPGS	962	949	-13	-1.35%
Paging				
READS/SEC	82	81	-1	-1.22%
WRITES/SEC	67	66	-1	-1.49%
PAGE/CMD	14.765	14.555	-0.211	-1.43%
PAGE IO RATE (V)	25.900	25.300	-0.600	-2.32%
PAGE IO/CMD (V)	2.567	2.505	-0.062	-2.40%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.441	6.436	-0.005	-0.08%

Table 5 (Page 2 of 2). Minidisk-only CMS-intensive migration from VM/ESA Release 2 on the 9221-170				
Release **Run ID**	**VM/ESA 2** **H15E0290**	**VM/ESA 2.1** **H16E0290**	**Difference**	**% Difference**
Environment				
Real Storage	**48MB**	**48MB**		
Exp. Storage	**16MB**	**16MB**		
Users	**290**	**290**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**1**	**1**		
Queues				
DISPATCH LIST	10.93	9.91	-1.02	-9.30%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	108	108	0	0.00%
VIO/CMD	10.702	10.693	-0.009	-0.08%
RIO RATE (V)	70	70	0	0.00%
RIO/CMD (V)	6.937	6.931	-0.006	-0.08%
MDC READS	56	55	-1	-1.79%
MDC WRITES	27	27	0	0.00%
MDC MODS	22	22	0	0.00%
MDC HIT RATIO	0.91	0.91	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	14.678	14.639	-0.039	-0.27%
DIAG/CMD	32.039	31.949	-0.090	-0.28%
DIAG 08/CMD	0.779	0.758	-0.021	-2.70%
DIAG 14/CMD	0.025	0.024	0.000	-0.92%
DIAG 58/CMD	1.253	1.254	0.001	0.07%
DIAG 98/CMD	2.343	2.344	0.001	0.03%
DIAG A4/CMD	3.936	3.906	-0.030	-0.75%
DIAG A8/CMD	1.883	1.883	-0.001	-0.04%
DIAG 214/CMD	13.036	13.001	-0.036	-0.27%
SIE/CMD	60.746	60.497	-0.250	-0.41%
SIE INTCPT/CMD	44.345	44.163	-0.182	-0.41%
FREE TOTL/CMD	74.422	74.854	0.432	0.58%
VTAM Machines				
WKSET (V)	296	295	-1	-0.34%
TOT CPU/CMD (V)	17.6723	17.2447	-0.4276	-2.42%
CP CPU/CMD (V)	7.7075	7.3160	-0.3915	-5.08%
VIRT CPU/CMD (V)	9.9647	9.9288	-0.0359	-0.36%
DIAG 98/CMD (V)	2.342	2.342	-0.001	-0.04%
Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM				

OfficeVision

The following sections document the performance measurements made in an OfficeVision (OV/VM) environment on a 9121-480 and a 9121-742.

The results of migrating from VM/ESA Release 2 to VM/ESA Release 2.1 showed an improvement for the environments measured, as shown in the following 2 graphs.

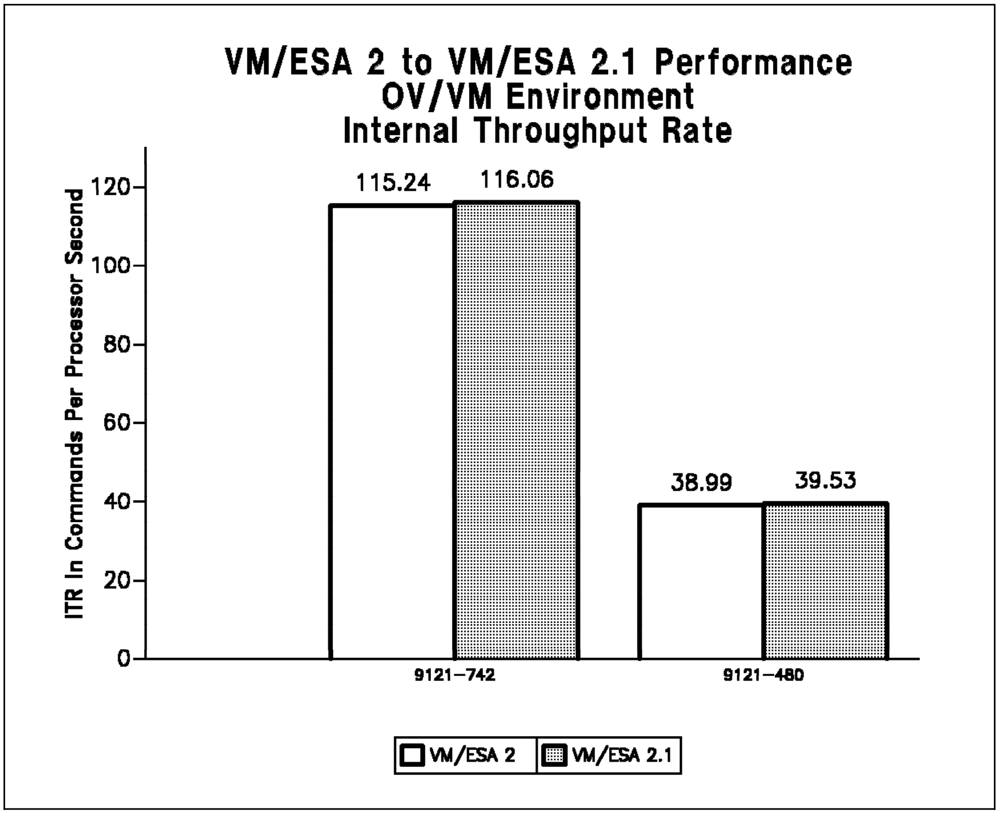

Figure 3. Internal throughput rate for OfficeVision/VM*

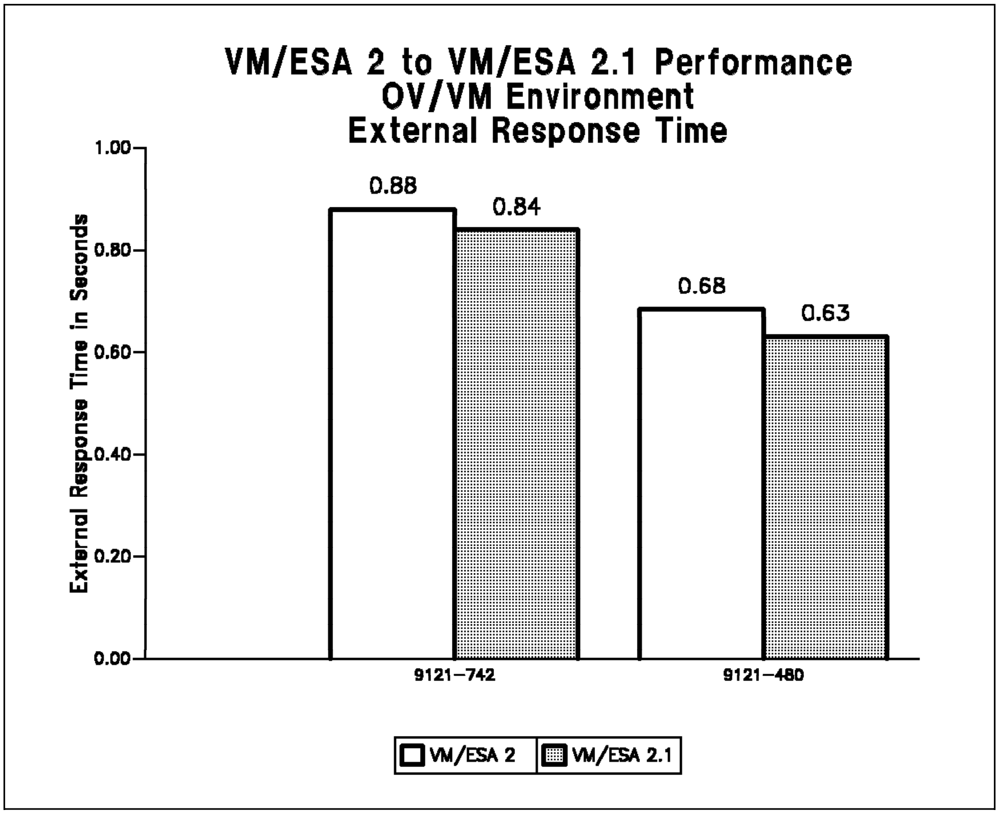

Figure 4. External response time for OfficeVision/VM

9121-742

Workload: IOB

Hardware Configuration

Processor model:	9121-742
Processors used:	4
Storage	
Real:	1024MB
Expanded:	1024MB (64MB to 1024MB for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-3	2	6	5				
3390-2	3990-2	4	16	6	6			
3390-2	3990-3	4		2	7	32 W	12 W	2 R

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	IOB
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
Ennnn	20	Workload	2MB/XA	7000		QUICKDSP ON
PRNTEAT1	1	Workload	2MB/XA	7000		QUICKDSP ON
PROCAL	1	OV/VM	16MB/XA	7000	1650	QUICKDSP ON
PRODBM	1	OV/VM	16MB/XA	7000	1000	QUICKDSP ON
PROMAIL	1	OV/VM	16MB/XA	7000		QUICKDSP ON
PROMBX	1	OV/VM	16MB/XA	7000		QUICKDSP ON IBCENTRL = Y
PROMBXnn	20	OV/VM	16MB/XA	7000		QUICKDSP ON IBCENTRL = Y
SMART	1	RTM	16MB/370	3%	400	QUICKDSP ON
VMCF	1	Monitor	4MB/XA	200		QUICKDSP ON
VSCSn	4	VSCS	64MB/XA	10000	900	QUICKDSP ON
VTAMXA	1	VTAM	64MB/XA	10000	550	QUICKDSP ON
WRITER	1	CP Monitor	2MB/XA	100		QUICKDSP ON
Users	6100	User	2MB/XA	100		

Note: *IBCENTRL = Y* is an OV/VM option causing the users′ inbaskets to reside in the mail box machines and not on the users′ A-disks for convenience of workload setup.

Measurement Discussion: This section documents the migration data collected for an OV/VM Release 2 environment running on a 9121-742. The data shown compare VM/ESA Release 2 and VM/ESA Release 2.1 in the environment described above. DASD fast write was enabled for these measurements due to the limited amount of DASD available in this hardware configuration.

The 9121-742 experienced a 0.7% increase in internal throughput rate (ITR (H)) and a 4.6% decrease in external response time (AVG LAST (T)). These improvements can be attributed to the load wait state PSW and IUCV processor usage items described in "Performance Improvements" on page 10.

Table 6 (Page 1 of 3). OfficeVision migration from VM/ESA Release 2 on the 9121-742

Release **Run ID**	**VM/ESA 2** **S45V6106**	**VM/ESA 2.1** **S46V6102**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**6100**	**6100**		
VTAMs	**1**	**1**		
VSCSs	**4**	**4**		
Processors	**4**	**4**		
Response Time				
TRIV INT	0.063	0.061	-0.002	-3.17%
NONTRIV INT	0.434	0.420	-0.014	-3.23%
TOT INT	0.391	0.388	-0.003	-0.77%
TOT INT ADJ	0.403	0.385	-0.018	-4.46%
AVG FIRST (T)	0.549	0.523	-0.027	-4.91%
AVG LAST (T)	0.880	0.840	-0.040	-4.57%
Throughput				
AVG THINK (T)	46.18	46.34	0.16	0.34%
ETR	106.39	102.39	-4.00	-3.76%
ETR (T)	103.23	103.20	-0.04	-0.04%
ETR RATIO	1.031	0.992	-0.038	-3.73%
ITR (H)	115.24	116.06	0.82	0.72%
ITR	29.72	28.83	-0.89	-3.01%
EMUL ITR	54.61	53.34	-1.27	-2.33%
ITRR (H)	1.000	1.007	0.007	0.72%
ITRR	1.000	0.970	-0.030	-3.01%
Proc. Usage				
PBT/CMD (H)	34.711	34.464	-0.247	-0.71%
PBT/CMD	34.679	34.497	-0.181	-0.52%
CP/CMD (H)	17.128	16.854	-0.273	-1.60%
CP/CMD	15.790	15.892	0.103	0.65%
EMUL/CMD (H)	17.583	17.610	0.027	0.15%
EMUL/CMD	18.889	18.605	-0.284	-1.50%

Table 6 (Page 2 of 3). OfficeVision migration from VM/ESA Release 2 on the 9121-742				
Release **Run ID**	**VM/ESA 2** **S45V6106**	**VM/ESA 2.1** **S46V6102**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**6100**	**6100**		
VTAMs	**1**	**1**		
VSCSs	**4**	**4**		
Processors	**4**	**4**		
Processor Util.				
TOTAL (H)	358.33	355.66	-2.67	-0.75%
TOTAL	358.00	356.00	-2.00	-0.56%
UTIL/PROC (H)	89.58	88.92	-0.67	-0.75%
UTIL/PROC	89.50	89.00	-0.50	-0.56%
TOTAL EMUL (H)	181.52	181.73	0.21	0.12%
TOTAL EMUL	195.00	192.00	-3.00	-1.54%
MASTER TOTAL (H)	94.46	93.95	-0.50	-0.53%
MASTER TOTAL	94.00	94.00	0.00	0.00%
MASTER EMUL (H)	20.03	20.43	0.40	1.99%
MASTER EMUL	22.00	22.00	0.00	0.00%
TVR(H)	1.97	1.96	-0.02	-0.86%
TVR	1.84	1.85	0.02	1.00%
Storage				
NUCLEUS SIZE (V)	2304KB	2368KB	64KB	2.78%
TRACE TABLE (V)	800KB	800KB	0KB	0.00%
WKSET (V)	85	83	-2	-2.35%
PGBLPGS	237 k	236 k	-1 k	-0.42%
PGBLPGS/USER	38.9	38.7	-0.2	-0.42%
FREEPGS	15277	15680	403	2.64%
FREE UTIL	0.96	0.96	0.01	0.85%
SHRPGS	1820	1780	-40	-2.20%
Paging				
READS/SEC	660	632	-28	-4.24%
WRITES/SEC	635	618	-17	-2.68%
PAGE/CMD	12.544	12.113	-0.432	-3.44%
PAGE IO RATE (V)	212.400	202.800	-9.600	-4.52%
PAGE IO/CMD (V)	2.057	1.965	-0.092	-4.49%
XSTOR IN/SEC	380	408	28	7.37%
XSTOR OUT/SEC	1061	1055	-6	-0.57%
XSTOR/CMD	13.959	14.177	0.218	1.56%
FAST CLR/CMD	15.644	15.640	-0.004	-0.03%
Queues				
DISPATCH LIST	108.51	102.63	-5.88	-5.42%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	2306	2314	8	0.35%
VIO/CMD	22.338	22.423	0.085	0.38%
RIO RATE (V)	1080	1076	-4	-0.37%
RIO/CMD (V)	10.462	10.427	-0.035	-0.33%
MDC READS	1151	1155	4	0.35%
MDC WRITES	1097	1102	5	0.46%
MDC MODS	928	933	5	0.54%
MDC HIT RATIO	0.87	0.87	0.00	0.00%

Table 6 (Page 3 of 3). OfficeVision migration from VM/ESA Release 2 on the 9121-742				
Release **Run ID**	**VM/ESA 2** **S45V6106**	**VM/ESA 2.1** **S46V6102**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**6100**	**6100**		
VTAMs	**1**	**1**		
VSCSs	**4**	**4**		
Processors	**4**	**4**		
PRIVOPs				
PRIVOP/CMD	27.037	27.769	0.733	2.71%
DIAG/CMD	84.309	85.510	1.201	1.42%
DIAG 08/CMD	10.668	10.708	0.039	0.37%
DIAG 14/CMD	1.044	1.047	0.003	0.31%
DIAG 58/CMD	2.056	2.057	0.000	0.02%
DIAG 98/CMD	0.550	0.552	0.002	0.27%
DIAG A4/CMD	10.828	10.868	0.039	0.36%
DIAG A8/CMD	6.134	6.126	-0.008	-0.13%
DIAG 214/CMD	33.392	34.467	1.075	3.22%
SIE/CMD	116.242	116.283	0.041	0.04%
SIE INTCPT/CMD	84.856	84.887	0.030	0.04%
FREE TOTL/CMD	174.363	174.425	0.062	0.04%
VTAM Machines				
WKSET (V)	3726	3702	-24	-0.64%
TOT CPU/CMD (V)	3.8995	3.7074	-0.1921	-4.93%
CP CPU/CMD (V)	2.0008	1.8537	-0.1471	-7.35%
VIRT CPU/CMD (V)	1.8986	1.8537	-0.0449	-2.36%
DIAG 98/CMD (V)	0.549	0.552	0.003	0.48%
Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM				

9121-480

Workload: IOB

Hardware Configuration

Processor model: 9121-480
Processors used: 2
Storage
  Real: 224MB
  Expanded: 32MB (all for MDC)
Tape: 3480 (Monitor)

DASD:

Type of DASD	Control Unit	Number of Paths	PAGE	SPOOL	- Number of Volumes - TDSK	User	Server	System
3390-2	3990-2	4	6	4	6			
3390-2	3990-3	2						2 R
3390-2	3990-3	4	10	4	2	16 R	7 R	

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	Number	Lines per Control Unit	Speed
3088-08	1	NA	4.5MB

Software Configuration

Driver: TPNS
Think time distribution: IOB
CMS block size: 4KB

Virtual Machines:

Virtual Machine	Number	Type	Machine Size/Mode	SHARE	RESERVED	Other Options
Ennnn	10	Workload	2MB/XA	1000		QUICKDSP ON
PRNTEAT1	1	Workload	2MB/XA	1000		QUICKDSP ON
PROCAL	1	OV/VM	16MB/XA	3000	1600	QUICKDSP ON
PRODBM	1	OV/VM	16MB/XA	3000	550	QUICKDSP ON
PROMAIL	1	OV/VM	16MB/XA	3000		QUICKDSP ON
PROMBX	1	OV/VM	16MB/XA	3000		QUICKDSP ON IBCENTRL = Y
PROMBXnn	10	OV/VM	16MB/XA	3000		QUICKDSP ON IBCENTRL = Y
SMART	1	RTM	16MB/370	3%	400	QUICKDSP ON
VMCF	1	Monitor	4MB/XA	200		QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	900	QUICKDSP ON
WRITER	1	CP Monitor	2MB/XA	100		QUICKDSP ON
Users	2100	User	2MB/XA	100		

Note: *IBCENTRL = Y* is an OV/VM option causing the users′ inbaskets to reside in the mail box machines and not on the users′ A-disks for convenience of workload setup.

Measurement Discussion: This section documents the migration data collected for an OV/VM Release 2 environment running on a 9121-480. The data shown compare VM/ESA Release 2 and VM/ESA Release 2.1 in the environment described above.

The 9121-480 experienced a 1.4% increase in internal throughput rate (ITR (H)) and a 7.8% decrease in external response time (AVG LAST (T)). These improvements can be attributed to the load wait state PSW and IUCV processor usage items described in "Performance Improvements" on page 10.

Table 7 (Page 1 of 3). OfficeVision migration from VM/ESA Release 2 on the 9121-480

Release **Run ID**	**VM/ESA 2** **L25V2103**	**VM/ESA 2.1** **L26V2102**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**2100**	**2100**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.056	0.058	0.002	3.57%
NONTRIV INT	0.660	0.606	-0.054	-8.18%
TOT INT	0.637	0.585	-0.052	-8.16%
TOT INT ADJ	0.593	0.544	-0.049	-8.28%
AVG FIRST (T)	0.463	0.432	-0.031	-6.80%
AVG LAST (T)	0.684	0.631	-0.053	-7.75%
Throughput				
AVG THINK (T)	46.39	46.40	0.01	0.01%
ETR	33.04	33.02	-0.02	-0.06%
ETR (T)	35.51	35.54	0.02	0.07%
ETR RATIO	0.930	0.929	-0.001	-0.13%
ITR (H)	38.99	39.53	0.54	1.38%
ITR	18.14	18.38	0.24	1.30%
EMUL ITR	32.70	33.07	0.37	1.12%
ITRR (H)	1.000	1.014	0.014	1.38%
ITRR	1.000	1.013	0.013	1.30%
Proc. Usage				
PBT/CMD (H)	51.289	50.592	-0.698	-1.36%
PBT/CMD	51.249	50.653	-0.596	-1.16%
CP/CMD (H)	24.614	23.973	-0.641	-2.60%
CP/CMD	22.809	22.512	-0.296	-1.30%
EMUL/CMD (H)	26.675	26.618	-0.057	-0.21%
EMUL/CMD	28.440	28.141	-0.300	-1.05%

Table 7 (Page 2 of 3). OfficeVision migration from VM/ESA Release 2 on the 9121-480

Release Run ID	VM/ESA 2 L25V2103	VM/ESA 2.1 L26V2102	Difference	% Difference
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**2100**	**2100**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Processor Util.				
TOTAL (H)	182.14	179.78	-2.36	-1.30%
TOTAL	182.00	180.00	-2.00	-1.10%
UTIL/PROC (H)	91.07	89.89	-1.18	-1.30%
UTIL/PROC	91.00	90.00	-1.00	-1.10%
TOTAL EMUL (H)	94.73	94.59	-0.14	-0.15%
TOTAL EMUL	101.00	100.00	-1.00	-0.99%
MASTER TOTAL (H)	91.59	90.44	-1.15	-1.26%
MASTER TOTAL	92.00	90.00	-2.00	-2.17%
MASTER EMUL (H)	37.56	37.48	-0.07	-0.20%
MASTER EMUL	40.00	40.00	0.00	0.00%
TVR(H)	1.92	1.90	-0.02	-1.15%
TVR	1.80	1.80	0.00	-0.11%
Storage				
NUCLEUS SIZE (V)	2300KB	2364KB	64KB	2.78%
TRACE TABLE (V)	400KB	400KB	0KB	0.00%
WKSET (V)	85	84	-1	-1.18%
PGBLPGS	47948	47814	-134	-0.28%
PGBLPGS/USER	22.8	22.8	-0.1	-0.28%
FREEPGS	5256	5394	138	2.63%
FREE UTIL	0.97	1.00	0.02	2.31%
SHRPGS	1327	1309	-18	-1.36%
Paging				
READS/SEC	624	622	-2	-0.32%
WRITES/SEC	574	575	1	0.17%
PAGE/CMD	33.734	33.684	-0.050	-0.15%
PAGE IO RATE (V)	161.700	158.500	-3.200	-1.98%
PAGE IO/CMD (V)	4.553	4.460	-0.093	-2.04%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	16.557	16.547	-0.011	-0.07%
Queues				
DISPATCH LIST	44.36	41.15	-3.21	-7.23%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	866	867	1	0.12%
VIO/CMD	24.386	24.398	0.012	0.05%
RIO RATE (V)	512	509	-3	-0.59%
RIO/CMD (V)	14.417	14.324	-0.094	-0.65%
MDC READS	408	408	0	0.00%
MDC WRITES	385	385	0	0.00%
MDC MODS	332	332	0	0.00%
MDC HIT RATIO	0.88	0.88	0.00	0.00%

Table 7 (Page 3 of 3). OfficeVision migration from VM/ESA Release 2 on the 9121-480				
Release **Run ID**	**VM/ESA 2** **L25V2103**	**VM/ESA 2.1** **L26V2102**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**2100**	**2100**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
PRIVOPs				
PRIVOP/CMD	21.151	21.616	0.465	2.20%
DIAG/CMD	92.725	93.389	0.665	0.72%
DIAG 08/CMD	11.896	11.917	0.021	0.18%
DIAG 14/CMD	1.504	1.509	0.004	0.28%
DIAG 58/CMD	2.058	2.060	0.001	0.06%
DIAG 98/CMD	1.388	1.395	0.007	0.47%
DIAG A4/CMD	11.128	11.117	-0.011	-0.10%
DIAG A8/CMD	6.320	6.332	0.012	0.20%
DIAG 214/CMD	35.413	36.054	0.640	1.81%
SIE/CMD	125.504	125.704	0.200	0.16%
SIE INTCPT/CMD	90.363	90.507	0.144	0.16%
FREE TOTL/CMD	190.776	189.132	-1.644	-0.86%
VTAM Machines				
WKSET (V)	900	900	0	0.00%
TOT CPU/CMD (V)	4.9591	4.7057	-0.2534	-5.11%
CP CPU/CMD (V)	2.2058	2.0324	-0.1734	-7.86%
VIRT CPU/CMD (V)	2.7533	2.6733	-0.0800	-2.91%
DIAG 98/CMD (V)	1.389	1.395	0.006	0.44%

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

VSE/ESA Guest

This section examines VSE/ESA 1.2.0 guest performance running under VM/ESA Release 2.1 compared to VM/ESA Release 2 guest performance. The PACEX8 workload was used for all measurements. PACEX8 is batch-only and is characterized by heavy I/O. See Appendix B, "Workloads" on page 191 for a detailed description of the workload.

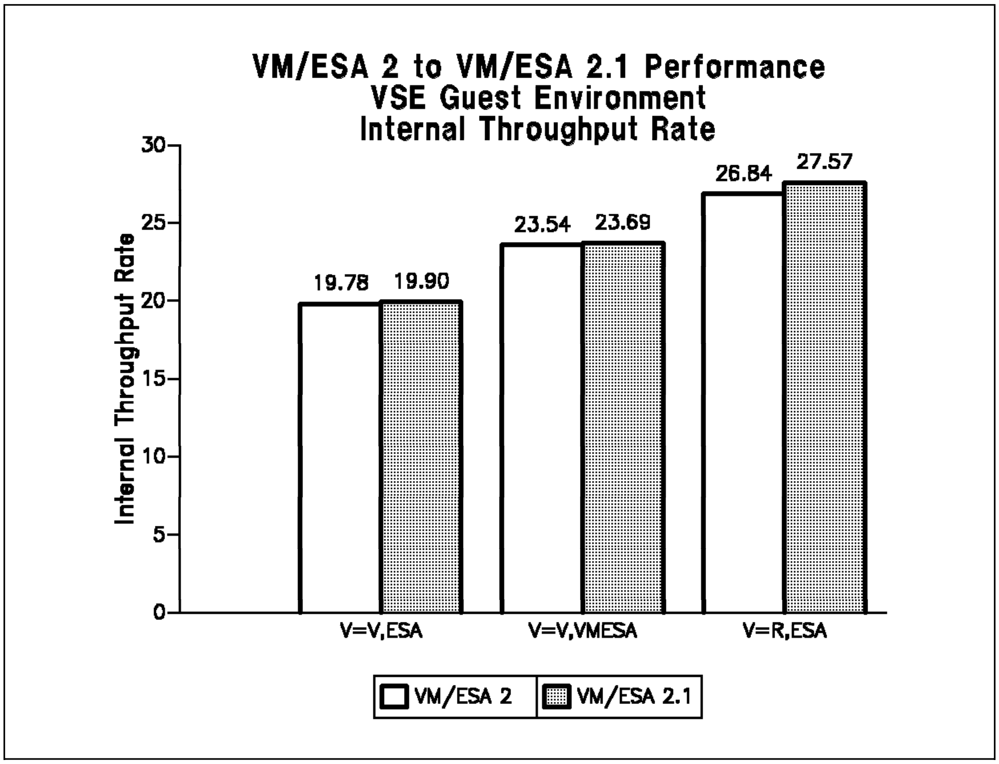

Figure 5. VSE guest migration from VM/ESA Release 2. PACEX8 workload on a single VSE/ESA* guest of VM/ESA Release 2 and VM/ESA Release 2.1 on the 9121-320 processor.

The PACEX8 workload consists of the base PACEX1 workload (7 batch jobs), running eight copies in eight partitions. All partitions are balanced with equal priorities. For all guest measurements in this section, VSE/ESA was run in an ESA virtual machine and the VSE supervisor was defined as MODE=ESA.

The VSE PACEX8 workload was run as a guest of VM/ESA Release 2 and VM/ESA Release 2.1 in three modes. The V=V guest results for VM/ESA Release 2.1 improved by about 0.6% in the internal throughput rate compared to VM/ESA Release 2. The V=R environment for VM/ESA Release 2.1 improved by 2.7% in the internal throughput rate compared to the same environment running under VM/ESA Release 2. This is mainly attributable to the load PSW wait state improvement in VM/ESA Release 2.1.

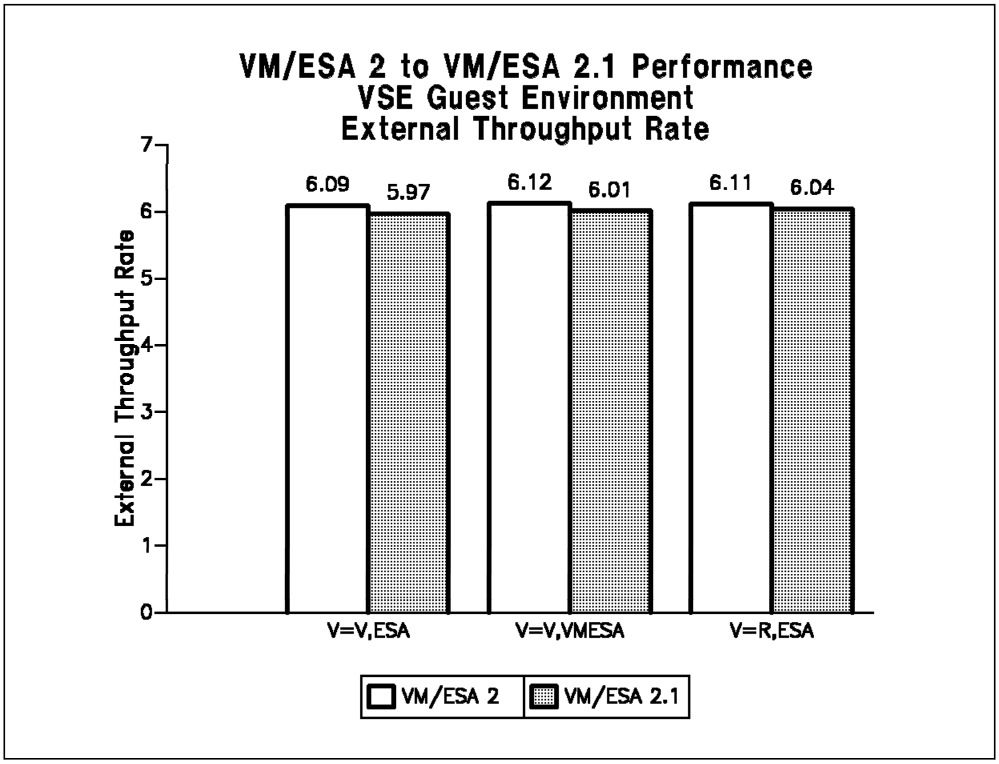

Figure 6. VSE guest migration from VM/ESA Release 2. PACEX8 workload on a single VSE/ESA guest of VM/ESA Release 2 and VM/ESA Release 2.1 on the 9121-320 processor.

The external throughput rate chart reflects elapsed time comparisons of the various guest modes running under VM/ESA Release 2 and VM/ESA Release 2.1. The elapsed time duration of the batch jobs remains relatively unchanged for all environments.

Workload: PACEX8

Hardware Configuration

Processor models: 9221-170, 9121-320[2]
Storage
  Real: 128MB, 256MB respectively
  Expanded: 0MB, 64MB respectively
Tape: 3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*VSAM*	*VSE Sys.*	*VM Sys.*
3380-A	3880-03	2	1			10	2	4

Software Configuration

VSE version: 1.2.0

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
USERVR	1	VSE V=R	24MB/ESA	100		IOASSIST ON CCWTRANS OFF
or						
USERV1	1	VSE V=V	24MB/ESA	100		IOASSIST OFF
SMART	1	RTM	16MB/370	100		
SETUP	1	CP monitor	2MB/XA	100		

[2] See “Hardware Used” on page 18 for an explanation of how this processor model was defined.

9121-320 / V=V MODE=ESA

Table 8. VSE/ESA guest migration from VM/ESA Release 2 on the 9121-320.

Release Run ID	VM/ESA 2 LB58VEX7	VM/ESA 2.1 LB68VEX9	Difference	% Difference
Environment				
Real Storage	**192MB**	**192MB**		
Expanded Storage	**64MB**	**64MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=V**	**V=V**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
Throughput (Min)				
Elapsed Time (H)	552.0	563.0	11.0	1.99%
ETR (H)	6.09	5.97	-0.12	-1.95%
ITR (H)	19.78	19.90	0.12	0.62%
ITR	19.64	19.89	0.26	1.31%
ITRR (H)	1.000	1.006	0.006	0.62%
ITRR	1.000	1.013	0.013	1.31%
Proc. Usage (Sec)				
PBT/CMD (H)	3.033	3.014	-0.019	-0.62%
PBT/CMD	3.056	3.016	-0.040	-1.30%
CP/CMD (H)	1.093	1.070	-0.023	-2.13%
CP/CMD	0.986	1.005	0.020	1.99%
EMUL/CMD (H)	1.940	1.944	0.004	0.23%
EMUL/CMD	2.070	2.011	-0.059	-2.86%
Processor Util.				
TOTAL (H)	30.77	29.98	-0.79	-2.56%
TOTAL	31.00	30.00	-1.00	-3.23%
TOTAL EMUL (H)	19.68	19.34	-0.34	-1.73%
TOTAL EMUL	21.00	20.00	-1.00	-4.76%
TVR(H)	1.56	1.55	-0.01	-0.85%
TVR	1.48	1.50	0.02	1.61%
Storage				
NUCLEUS SIZE (V)	2012KB	2076KB	64KB	3.18%
TRACE TABLE (V)	64KB	64KB	0KB	0.00%
PGBLPGS	15058	15039	-19	-0.13%
FREEPGS	80	81	1	1.25%
FREE UTIL	0.53	0.53	0.01	1.12%
SHRPGS	930	913	-17	-1.83%
Paging				
PAGE/CMD	0.000	0.000	0.000	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	88.714	90.482	1.768	1.99%
I/O				
VIO RATE	317.000	311.000	-6.000	-1.89%
VIO/CMD	3124.714	3126.661	1.946	0.06%
RIO RATE (V)	315.000	314.000	-1.000	-0.32%
RIO/CMD (V)	3105.000	3156.821	51.821	1.67%
DASD IO TOTAL (V)	169933	169059	-874	-0.51%
DASD IO RATE (V)	314.69	313.07	-1.62	-0.51%
DASD IO/CMD (V)	3101.95	3147.49	45.54	1.47%
PRIVOPs				
PRIVOP/CMD	3125.181	3124.722	-0.459	-0.01%
DIAG/CMD	327.656	326.077	-1.579	-0.48%
SIE/CMD	10428.857	10425.554	-3.304	-0.03%
SIE INTCPT/CMD	9385.971	9382.998	-2.973	-0.03%
FREE TOTL/CMD	4011.857	3800.250	-211.607	-5.27%

Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM

9121-320 / V=V MODE=VMESA

Table 9. VSE/ESA guest migration from VM/ESA Release 2 on the 9121-320.

Release Run ID	VM/ESA 2 LB58VEM5	VM/ESA 2.1 LB68VEMA	Difference	% Difference
Environment				
Real Storage	**192MB**	**192MB**		
Expanded Storage	**64MB**	**64MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=V**	**V=V**		
VSE Supervisor	**VMESA**	**VMESA**		
Processors	**1**	**1**		
Throughput (Min)				
Elapsed Time (H)	549.0	559.0	10.0	1.82%
ETR (H)	6.12	6.01	-0.11	-1.79%
ITR (H)	23.54	23.69	0.15	0.64%
ITR	29.14	26.13	-3.01	-10.33%
ITRR (H)	1.000	1.006	0.006	0.64%
ITRR	1.000	0.897	-0.103	-10.33%
Proc. Usage (Sec)				
PBT/CMD (H)	2.549	2.533	-0.016	-0.64%
PBT/CMD	2.059	2.296	0.237	11.52%
CP/CMD (H)	1.006	0.998	-0.008	-0.80%
CP/CMD	0.784	0.799	0.014	1.82%
EMUL/CMD (H)	1.543	1.535	-0.008	-0.53%
EMUL/CMD	1.274	1.497	0.223	17.49%
Processor Util.				
TOTAL (H)	26.00	25.37	-0.63	-2.42%
TOTAL	21.00	23.00	2.00	9.52%
TOTAL EMUL (H)	15.74	15.38	-0.36	-2.31%
TOTAL EMUL	13.00	15.00	2.00	15.38%
TVR(H)	1.65	1.65	0.00	-0.11%
TVR	1.62	1.53	-0.08	-5.08%
Storage				
NUCLEUS SIZE (V)	2012KB	2076KB	64KB	3.18%
TRACE TABLE (V)	64KB	64KB	0KB	0.00%
PGBLPGS	15059	15040	-19	-0.13%
FREEPGS	79	80	1	1.27%
FREE UTIL	0.53	0.54	0.01	1.10%
SHRPGS	930	913	-17	-1.83%
Paging				
PAGE/CMD	0.000	0.000	0.000	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	68.625	79.857	11.232	16.37%
I/O				
VIO RATE	261.000	282.000	21.000	8.05%
VIO/CMD	2558.732	2814.964	256.232	10.01%
RIO RATE (V)	321.000	290.000	-31.000	-9.66%
RIO/CMD (V)	3146.946	2894.821	-252.125	-8.01%
DASD IO TOTAL (V)	172972	173532	560	0.32%
DASD IO RATE (V)	320.32	289.22	-31.10	-9.71%
DASD IO/CMD (V)	3140.27	2887.04	-253.23	-8.06%
PRIVOPs				
PRIVOP/CMD	2558.952	2820.814	261.862	10.23%
DIAG/CMD	145.072	146.345	1.273	0.88%
SIE/CMD	8264.411	9103.714	839.304	10.16%
SIE INTCPT/CMD	7603.258	8375.417	772.159	10.16%
FREE TOTL/CMD	3244.982	3443.839	198.857	6.13%

Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM

9121-320 / V=R MODE=ESA

Table 10. VSE/ESA guest migration from VM/ESA Release 2 on the 9121-320.

Release Run ID	VM/ESA 2 LB58REX5	VM/ESA 2.1 LB68REXB	Difference	% Difference
Environment				
Real Storage	**192MB**	**192MB**		
Expanded Storage	**64MB**	**64MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=R**	**V=R**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
Throughput (Min)				
Elapsed Time (H)	550.0	556.0	6.0	1.09%
ETR (H)	6.11	6.04	-0.07	-1.08%
ITR (H)	26.84	27.57	0.73	2.73%
ITR	26.56	27.47	0.91	3.42%
ITRR (H)	1.000	1.027	0.027	2.73%
ITRR	1.000	1.034	0.034	3.42%
Proc. Usage (Sec)				
PBT/CMD (H)	2.235	2.176	-0.060	-2.66%
PBT/CMD	2.259	2.184	-0.075	-3.30%
CP/CMD (H)	0.381	0.327	-0.055	-14.31%
CP/CMD	0.393	0.298	-0.095	-24.18%
EMUL/CMD (H)	1.854	1.849	-0.005	-0.27%
EMUL/CMD	1.866	1.886	0.020	1.09%
Processor Util.				
TOTAL (H)	22.76	21.92	-0.85	-3.71%
TOTAL	23.00	22.00	-1.00	-4.35%
TOTAL EMUL (H)	18.88	18.63	-0.25	-1.35%
TOTAL EMUL	19.00	19.00	0.00	0.00%
TVR(H)	1.21	1.18	-0.03	-2.40%
TVR	1.21	1.16	-0.05	-4.35%
Storage				
NUCLEUS SIZE (V)	2012KB	2076KB	64KB	3.18%
TRACE TABLE (V)	64KB	64KB	0KB	0.00%
PGBLPGS	15067	15038	-29	-0.19%
FREEPGS	78	80	2	2.56%
FREE UTIL	0.52	0.51	-0.01	-1.29%
SHRPGS	912	929	17	1.86%
Paging				
PAGE/CMD	0.000	0.000	0.000	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	0.000	0.000	0.000	na
I/O				
VIO RATE	5.000	5.000	0.000	0.00%
VIO/CMD	49.107	49.643	0.536	1.09%
RIO RATE (V)	1.000	1.000	0.000	0.00%
RIO/CMD (V)	9.821	9.929	0.107	1.09%
DASD IO TOTAL (V)	173080	171765	-1315	-0.76%
DASD IO RATE (V)	320.52	318.08	-2.44	-0.76%
DASD IO/CMD (V)	3147.95	3158.11	10.16	0.32%
PRIVOPs				
PRIVOP/CMD	57.200	56.483	-0.716	-1.25%
DIAG/CMD	322.360	324.088	1.728	0.54%
SIE/CMD	3201.786	3226.786	25.000	0.78%
SIE INTCPT/CMD	2849.589	2871.839	22.250	0.78%
FREE TOTL/CMD	883.929	873.714	-10.214	-1.16%

Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM

9221-170 / V=R MODE=ESA with Full Pack Minidisks

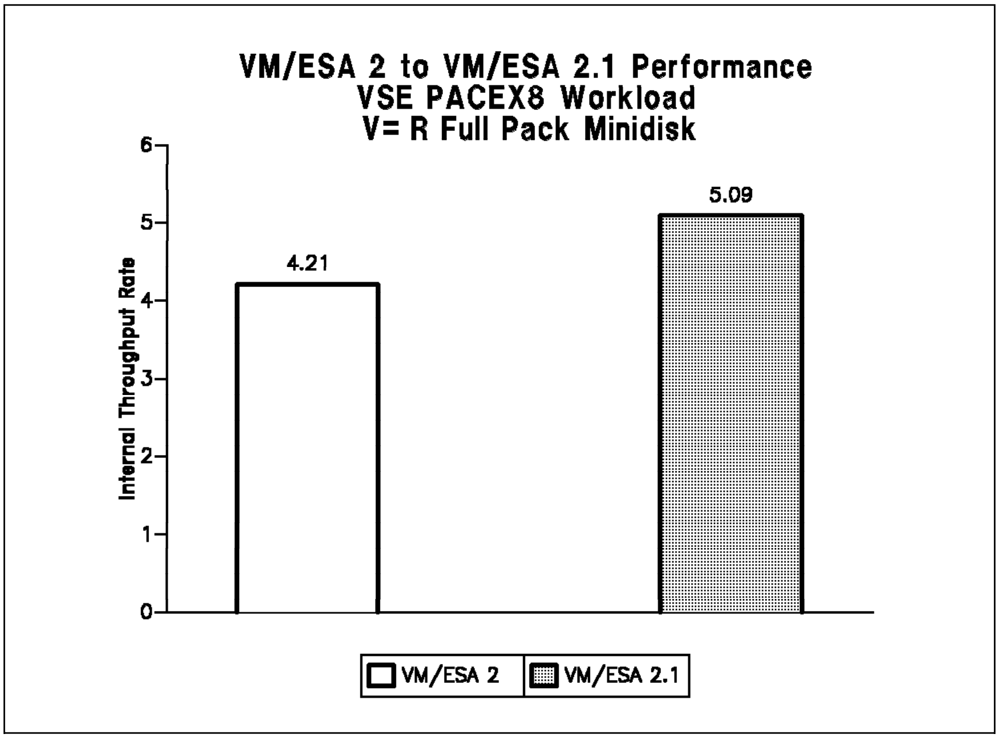

Figure 7. Full pack minidisk DASD configuration for V=R guest. PACEX8 workload on VSE/ESA under VM/ESA Release 2 and VM/ESA Release 2.1 on the 9221-170 processor.

The VSE batch workload was run as a V=R guest with DASD defined as full pack minidisks. In VM/ESA Release 2, this case did not receive the benefit of CCW fast path translation. VM/ESA Release 2.1 now incorporates CCW fast path translation for V=R guest machines with DASD defined as full pack minidisks. A 21% improvement in the internal throughput rate was achieved with this workload. A workload with less I/O content will not experience as much gain in internal throughput rate.

Table 11. VSE/ESA guest with DASD defined as full pack minidisks on the 9221-170.				
Release **Run ID**	**VM/ESA 2** **VR1170FE**	**VM/ESA 2.1** **VR1170FC**	**Difference**	**% Difference**
Environment				
IML Mode	**ESA**	**ESA**		
Real Storage	**128MB**	**128MB**		
Expanded Storage	**0MB**	**0MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=R**	**V=R**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
Throughput (Min)				
Elapsed Time (H)	930.1	741.5	-188.6	-20.28%
ETR (H)	3.61	4.53	0.92	25.44%
ITR (H)	4.21	5.09	0.88	21.03%
ITR	4.20	5.27	1.07	25.44%
ITRR (H)	1.000	1.210	0.210	21.03%
ITRR	1.000	1.254	0.254	25.44%
Proc. Usage (Sec)				
PBT/CMD (H)	14.262	11.784	-2.479	-17.38%
PBT/CMD	14.284	11.387	-2.897	-20.28%
CP/CMD (H)	7.960	5.484	-2.476	-31.10%
CP/CMD	6.810	4.105	-2.705	-39.72%
EMUL/CMD (H)	6.303	6.300	-0.003	-0.05%
EMUL/CMD	7.474	7.282	-0.192	-2.57%
Processor Util.				
TOTAL (H)	85.87	89.00	3.13	3.64%
TOTAL	86.00	86.00	0.00	0.00%
TOTAL EMUL (H)	37.95	47.58	9.63	25.38%
TOTAL EMUL	45.00	55.00	10.00	22.22%
TVR(H)	2.26	1.87	-0.39	-17.34%
TVR	1.91	1.56	-0.35	-18.18%
Storage				
NUCLEUS SIZE (V)	2012KB	2076KB	64KB	3.18%
TRACE TABLE (V)	128KB	64KB	-64KB	-50.00%
PGBLPGS	14625	14588	-37	-0.25%
FREEPGS	77	82	5	6.49%
FREE UTIL	0.52	0.54	0.02	3.88%
SHRPGS	929	912	-17	-1.83%
Paging				
PAGE/CMD	0.000	0.000	0.000	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	0.000	0.000	0.000	na
I/O				
VIO RATE	168.000	202.000	34.000	20.24%
VIO/CMD	2790.276	2674.557	-115.719	-4.15%
RIO RATE (V)	172.000	199.000	27.000	15.70%
RIO/CMD (V)	2856.712	2634.836	-221.876	-7.77%
DASD IO TOTAL (V)	170914	172635	1721	1.01%
DASD IO RATE (V)	189.90	221.33	31.42	16.55%
DASD IO/CMD (V)	3154.08	2930.45	-223.63	-7.09%
PRIVOPs				
PRIVOP/CMD	2790.259	2687.763	-102.496	-3.67%
DIAG/CMD	399.130	370.058	-29.072	-7.28%
SIE/CMD	9317.530	8963.738	-353.792	-3.80%
SIE INTCPT/CMD	7826.725	7260.628	-566.098	-7.23%
FREE TOTL/CMD	27138.760	6580.469	-20558.291	-75.75%

Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM

VMSES/E

A number of requirements were addressed in VM Service Enhancements Staged/Extended (VMSES/E) in VM/ESA Release 2.1. The following are new functions for this release:

- The VMFHLASM command has been added to support the use of the IBM High Level Assembler.
- Local modification support reduces the number of steps and eliminates redundancy when installing local support.
- The VMFBLD command automates the generation of callable services libraries and CMS/DOS phase libraries.
- The VMFBLD command automates the generation of generated objects, such as text decks.

A few performance enhancements were also included. For the VMFSIM function (which provides access to the VMSES/E Software Inventory), code restructure and the replacement of some REXX commands with pipelines code, reduced the time spent building a nucleus. Also, replacing REXX code with pipelines code in the BUILDLL routine for VMFBDNUC helped reduce the time spent building a nucleus.

These performance enhancements were offset slightly by code quality improvements that went into the build function. However, the performance results show that VMSES/E in VM/ESA Release 2.1 improved when compared to VMSES/E in VM/ESA Release 2.

Three primary VMSES/E tools that help with the servicing of products were measured to quantify the effects of the new function, performance enhancements, and code quality improvements:

- VMFREC EXEC receives the raw materials from a service tape and places them into the raw materials database.
- VMFAPPLY EXEC defines new maintenance levels based on the contents of the raw materials database.
- VMFBLD EXEC uses the defined maintenance levels to select the correct level and build the running product.

Overall, for the measurements reported here, the process of receiving and applying CMS service, and building CMS on VMSES/E in VM/ESA Release 2.1 improved total elapsed time (Total Time (Q)) when compared to VMSES/E in VM/ESA Release 2. On the 9121-480 configuration, the sum of the total elapsed time for the VMSES/E in VM/ESA Release 2.1 receive, apply, and build commands improved 2% when compared to VMSES/E in VM/ESA Release 2. Note that the overall improvement would have been greater if VMSES/E in VM/ESA Release 2 did not have APAR 54581 applied—performance enhancements for VMFSIMPC and VMFBDNUC.

Finally, for VMSES/E in VM/ESA Release 2.1, the savings in terms of new automation are a result of the new part handlers—VMFBDCLB and VMFBDDLB—for the generation of CSL and DOSLIB, respectively. Both of these new functions save manually researching what needs to be rebuilt. VMSES/E in VM/ESA Release 2.1 also provides full HLASM automation support. Because of the additional automation, the total time (including manual steps) required to complete these tasks can be significantly reduced.

The measurements described in the following sections are provided to demonstrate the performance of these changes on VMSES/E in VM/ESA Release 2.1.

Hardware Configuration

Processor model:	9121-480
Processors used:	2
Storage	
Real:	224MB
Expanded:	32MB
Tape:	3480 (one service tape for the receive command)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-3	2		1		1 R		2 R

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*RELSHARE*	*RESERVED*	*Other Options*
MAINT	1	MAINT	30MB/XA			

Measurement Discussion: All measurements were performed on a dedicated, first-level system with only one user logged on (the MAINT user ID). The objective of these measurements was to show that the new functional enhancements to VMSES/E in VM/ESA Release 2.1 did not degrade performance when compared to VMSES/E in VM/ESA Release 2 in an established service environment. That is, all Software Inventory Management (SIM) tables had been previously initialized using the same Recommended Service Upgrade (RSU) tape with both releases of VMSES/E. The purpose of initializing SIM was to remove the one-time costs associated with setting up SIM.

Once initialized, a Corrective (COR) service tape containing CMS service was loaded onto the system. The performance test system used for these measurements was set up so that the COR tape would be compatible with both VMSES/E in VM/ESA Release 2 and VMSES/E in VM/ESA Release 2.1. Hence, the VMSES/E function of both releases worked on exactly the same service and the same raw materials database.

The CMS service from the COR tape was received. VMFREC was used to receive a total of 1728 CMS parts from seven tape files. Next, the apply function (VMFAPPLY) was used to process 206 PTFs. The build function (VMFBLD) with the STATUS option was invoked and identified 149 build requirements. Finally, 15 build lists were processed after running the VMFBLD command with the SERVICED option.

The methodology described in this section applies to both VMSES/E in VM/ESA Release 2.1 and VMSES/E in VM/ESA Release 2. Performance data were collected before and after each command execution to determine total response time and the total amount of resources used by the execution of the command. The performance data were generated by the CP QUERY TIME command and CP IND USER command. No intermediate steps were necessary that required human intervention (for example, entering data, pressing a function key, or mounting a tape). Hence, the performance data reported were derived from uninterrupted running of the command.

The following performance indicators were used and can be found in the tables below:

Total Time (seconds): the total elapsed time for the command. This is computed by taking the difference between the start and stop time. More specifically, it is the time after the enter key is pressed (the command had already been typed) until the ready message is received.

Total CPU (seconds): the difference in TOTCPU for the user before and after running the command.

Virtual CPU (seconds): the difference in VIRTCPU for the user before and after running the command.

I/O (number): the difference in non-spooled I/O requests by the user before and after running the command.

Resident Pages growth (pages): the increase in the user′s real storage pages after command completion (compared to the start of running the command).

Working Set Size growth (pages): the increase in the user′s working set size after command completion (compared to the start of running the command).

Two performance factors were not included in the results: 1) the time taken to investigate the necessary steps to invoke the function and 2) the time to manually error check the correctness of the information or the results. (The successful completion of each service command was checked after the command finished.)

Workload: Receive

Command: VMFREC PPF ESA CMS
Scenario Details: 1728 parts received from 7 tape files.

Table 12. VMFREC measurement data: VMSES/E in VM/ESA Release 2 and VMSES/E in VM/ESA Release 2.1 on the 9121-480

Release	VMSES/E 2	VMSES/E 2.1	Difference	% Difference
Total Time (Q)	589	575	-14	-2%
Total CPU (Q)	202	186	-16	-8%
Virtual CPU (Q)	185	169	-16	-9%
I/O (I)	32993	33003	+10	+0%
Resident Pages Growth (I)	198	255	+57	+29%
Working Set Size Growth (I)	456	471	+15	+3%

Note: Q=CP QUERY TIME, I=CP INDICATE USER.

Workload: Apply

Command: VMFAPPLY PPF ESA CMS
Scenario Details: 206 PTFs after receiving parts from COR tape.

Table 13. VMFAPPLY measurement data: VMSES/E in VM/ESA Release 2 and VMSES/E in VM/ESA Release 2.1 on the 9121-480

Release	VMSES/E 2	VMSES/E 2.1	Difference	% Difference
Total Time (Q)	359	349	-10	-3%
Total CPU (Q)	293	283	-10	-3%
Virtual CPU (Q)	285	275	-10	-4%
I/O (I)	20500	20515	+15	+0%
Resident Pages Growth (I)	139	136	-3	-2%
Working Set Size Growth (I)	360	398	+38	+11%

Note: Q=CP QUERY TIME, I=CP INDICATE USER.

Workload: Build with STATUS Option

Command: VMFBLD PPF ESA CMS (STATUS
Scenario Details: 149 build requirements identified.

Table 14. VMFBLD STATUS measurement data: VMSES/E in VM/ESA Release 2 and VMSES/E in VM/ESA Release 2.1 on the 9121-480

Release	VMSES/E 2	VMSES/E 2.1	Difference	% Difference
Total Time (Q)	146	144	-2	-1%
Total CPU (Q)	134	132	-2	-2%
Virtual CPU (Q)	133	131	-2	-2%
I/O (I)	2528	2537	+9	0%
Resident Pages Growth (I)	840	927	+87	+10%
Working Set Size Growth (I)	949	786	-163	-17%

Note: Q=CP QUERY TIME, I=CP INDICATE USER.

Workload: Build with SERVICED Option

Command: VMFBLD PPF ESA CMS (SERVICED
Scenario Details: 16 build lists processed; 149 objects built.

Table 15. VMFBLD SERVICED measurement data: VMSES/E in VM/ESA Release 2 and VMSES/E in VM/ESA Release 2.1 on the 9121-480

Release	VMSES/E 2	VMSES/E 2.1	Difference	% Difference
Total Time (Q)	905	893	-12	-1%
Total CPU (Q)	546	540	-6	-1%
Virtual CPU (Q)	530	524	-6	-1%
I/O (I)	32177	32221	+44	0%
Resident Pages Growth (I)	852	989	+137	+16%
Working Set Size Growth (I)	847	1028	+181	+21%

Note: Q=CP QUERY TIME, I=CP INDICATE USER.

Migration from Other VM Releases

The performance results provided in this report apply to migration from VM/ESA Release 2. This section discusses how to use the information in this report along with similar information from earlier reports to get an understanding of the performance of migrating from earlier VM releases.

Migration Performance Measurements Matrix

The matrix on the following page is provided as an index to all the performance measurements pertaining to VM migration that are available in the VM/ESA performance reports. The numbers (1 through 4) that appear in the matrix indicate which report includes the migration results for that case:

1. *VM/ESA Release 1.0 Performance Report*, ZZ05-0469
2. *VM/ESA Release 1.1 Performance Report*, GG66-3236
3. *VM/ESA Release 2 Performance Report*, GG66-3245
4. *VM/ESA Release 2.1 Performance Report* (this document)

Many of the comparisons listed in the matrix are for two consecutive VM releases. For migrations that skip one or more VM releases, you can get a general idea how the migration will affect performance by studying the applicable results for those two or more comparisons that, in combination, span those VM releases. For example, to get a general understanding of how migrating from VM/ESA 1.1 to VM/ESA 2.1 will tend to affect VSE guest performance, look at the VM/ESA 1.1 to VM/ESA 2 comparison measurements and the VM/ESA 2 to VM/ESA 2.1 comparison measurements. In each case, use the measurements from the system configuration that best approximates your VM system. For more discussion on the use of multiple comparisons, see "Migration Summary: CMS-Intensive Environment" on page 70.

The comparisons listed for the CMS-intensive environment primarily consist of minidisk-only measurements but there are some SFS comparisons as well.

Internal throughput rate ratio (ITRR) information for the minidisk-only CMS-intensive environment has been extracted from the CMS comparisons listed in the matrix and is summarized in "Migration Summary: CMS-Intensive Environment" on page 70.

Table 16. Sources of VM migration performance measurement results

Source	Target	Processor	Report Number			
			CMS	OV/VM	VSE Guest	MVS Guest
VM/SP 5	VM/ESA 1.0 (370)	4381-13	1			
	VM/ESA 1.0 (370)	9221-170			3	
	VM/ESA 1.0 (370)	9221-120	3		3	
	VM/ESA 2	9221-170			3	
	VM/ESA 2	9221-120	3		3	
VM/SP 6	VM/ESA 1.0 (370)	4381-13	1			
		9370-80	1			
		9370-30	1			
VM/SP HPO 5	VM/ESA 1.0 (ESA)	3090-200J	1			
	VM/ESA 2	9121-480	3			
	VM/ESA 2	9121-320	3			
VM/ESA 1.0 (370)	VM/ESA 1.1	9221-170	2			
	VM/ESA 2	9221-170	3		3	
	VM/ESA 2	9221-120	3		3	
VM/XA 2.0	VM/ESA 1.0 (ESA)	3090-600J	1			
VM/XA 2.1	VM/ESA 1.0 (ESA)	3090-600J	1			1
	VM/ESA 1.0 (ESA)	3090-200J	1			
	VM/ESA 1.0 (ESA)	9021-720		2		
	VM/ESA 1.0 (ESA)	9121-320			2	
	VM/ESA 1.1	9021-720		2		
	VM/ESA 1.1	9121-320			2	
VM/ESA 1.0 (ESA)	VM/ESA 1.1	3090-600J				2
		9021-720	2	2		
		9021-580	2			
		9121-480	2			
		9121-320	2		2	
		9221-170	2			
VM/ESA 1.1	VM/ESA 2	9021-900	3			3
		9021-720		3		
		9121-480	3	3		
		9121-320			3	
		9221-170	3			
VM/ESA 2	VM/ESA 2.1	9121-742	4	4		
		9121-480	4	4		
		9121-320			4	
		9221-200			4	
		9221-170	4		4	

Migration Summary: CMS-Intensive Environment

A large body of performance information for the CMS-intensive environment was collected over the last several releases of VM. This section summarizes the internal throughput rate (ITR) data from those measurements to show, for CMS-intensive workloads, the approximate changes in processing capacity that may occur when migrating from one VM release to another. As such, this section can serve as one source of migration planning information.

The performance relationships shown here are limited to the minidisk-only CMS-intensive environment. Other types of VM usage may show different relationships. Furthermore, any one measure such as ITR cannot provide a complete picture of the performance differences between VM releases. The VM performance reports from which the ITR ratios (ITRRs) were extracted can serve as a good source of additional performance information. Those reports are listed on page 68.

Table 17 summarizes the ITR relationships that were observed for the CMS-intensive environment for a number of VM release-to-release transitions:

Table 17. Approximate VM relative capacity: CMS-intensive environment

Source	**Target**	**Case**	**ITRR**	**ITRR Derivation**	**Notes**
VM/SP 5	VM/ESA 1.0 (370) VM/ESA 2.1	 9221-120	0.92 0.84	R5 R5*R13a*R2	1,5 1,2,4,6
VM/SP 6	VM/ESA 1.0 (370) VM/ESA 2.1	 9221-120	1.07 0.97	R6 R6*R13a*R2	5 2,4,6
VM/ESA 1.0 (370)	VM/ESA 2.1	9221-120 9221-170	0.91 0.97	R13a*R2 R13b*R11*R2	2,6 4,5,6
VM/SP HPO 5	VM/ESA 2.1	UP, ¬4381 MP, ¬4381	0.92 1.03	RHa*R2 RHb*R1E*R11*R2	4,5 3,4,5
VM/XA 2.0	VM/ESA 2.1		1.12	RX20*RX21*R1E* R11*R2	
VM/XA 2.1	VM/ESA 2.1		1.10	RX21*R1E*R11*R2	
VM/ESA 1.0 (ESA)	VM/ESA 2.1		1.06	R1E*R11*R2	
VM/ESA 1.1	VM/ESA 2.1		1.02	R11*R2	
VM/ESA 2	VM/ESA 2.1		1.01	R2	

Explanation of columns:

Case The set of conditions for which the stated ITRR approximately applies. When not specified, no large variations in ITRR were found among the cases that were measured. However, smaller variations are typically seen. These ITRR variations are shown in "Derivation and Supporting Data" on page 72.

ITRR The target ITR divided by the source ITR. A number greater than 1.00 indicates an improvement in processor capacity.

ITRR Derivation Shows how the ITRR was derived. See "Derivation and Supporting Data" on page 72 for discussion.

Notes:

1. The VM/SP 5 system is assumed to include APAR VM30315, the performance SPE that adds segment protection and 4KB key support. Other measurements have shown that VM/SP 5 ITR is 4% to 6% lower without this APAR.

2. This includes an increase of central storage from 16MB to 32MB to compensate for VM/ESA′s larger storage requirements. The VM/ESA case also includes 16MB of expanded storage for minidisk caching.

3. The VM/SP HPO 5 to VM/ESA Release 1.0 (ESA Feature) portion of the derivation was done with a reduced think time to avoid a 16MB-line real storage constraint in the HPO case. In cases where the base HPO system is 16MB-line constrained, migration to VM/ESA will yield additional performance benefits by eliminating this constraint.

4. The ESA-capable 4381 models provide less processing capacity when run in ESA mode as compared to 370 mode. Therefore, expect a less favorable ITR ratio when migrating on a 4381 processor from VM/SP, VM/SP HPO, or VM/ESA Release 1.0 (370 Feature) to VM/ESA Release 2.1.

5. The target VM system supports a larger real memory size than the stated migration source and this potential benefit is not reflected in the stated ITR ratios. Migrations from memory-constrained environments will yield additional ITRR and other performance benefits when the target configuration has additional real storage.

   A VM/SP example: The stated VM/SP 5 to VM/ESA Release 1.0 (370 Feature) ITRR of 0.92 is based on measurements done on a system with 16MB of real memory. However, VM/ESA Release 1.0 (370 Feature) supports up to 64MB of real memory (but subject to the 16MB-line constraint). When VM/SP 5 with 16MB was compared to VM/ESA Release 1.0 (370 Feature) with 32MB, an ITRR of 0.98 was observed. See "CMS-Intensive Migration from VM/SP Release 5" in the *VM/ESA Release 2 Performance Report* for details.

   A VM/SP HPO example: The stated VM/SP HPO 5 to VM/ESA 2.1 ITRR (for uniprocessors) is based (in part) on a VM/SP HPO 5 to VM/ESA 2 comparison, which showed an ITRR of 0.91. Those measurements were done on a 9121-320 system with its 256MB of storage configured as 64MB of real storage and 192MB of expanded storage (64MB/192MB). The 9121-320 had to be configured that way because 64MB is the maximum real storage supported by HPO. When VM/SP HPO Release 5.0 (64MB/192MB) was compared to VM/ESA 2 (192MB/64MB), an ITRR of 0.95 was observed. See "CMS-Intensive Migration from VM/SP HPO Release 5" in the *VM/ESA Release 2 Performance Report* for details.

6. These results apply to the case where the following recommended tuning is done for the target system:

   - Configure 16MB as expanded storage for minidisk caching.

- On VM/ESA systems before VM/ESA Release 2, set DSPSLICE to three times the default. Otherwise, use the default value.
- For the 9221-120, set the VTAM DELAY operand in the VTAM CTCA channel-attachment major node to 0.3 seconds. For the 9221-170, set the VTAM delay to 0.2 seconds.
- Set IPOLL ON for VTAM.
- Preload the key shared segments.

See section "CMS-Intensive Migration from VM/ESA 1.1," subsection "9221-170 / Minidisk" in the *VM/ESA Release 2 Performance Report* for more information on these tuning items. The purpose of this tuning is to configure VM/ESA for use on ESA-mode 9221 processors. If this tuning is not done, lower ITR ratios will be experienced. For example, for the FS7B0R CMS-intensive workload, going from VM/ESA Release 1.0 (370 Feature) to VM/ESA 1.1 resulted in an ITRR of 0.95 with the above tuning and an ITRR of 0.86 without it. This comparison is shown in the *VM/ESA Release 1.1 Performance Report*.

This table only shows performance in terms of ITR ratios (processor capacity). It does not provide, for example, any response time information. An improved ITR tends to result in better response times and vice versa. However, exceptions occur. An especially noteworthy exception is the migration from 370-based VM releases to VM/ESA. In such migrations, response times have frequently been observed to improve significantly, even in the face of an ITR decrease. One pair of measurements, for example, showed a 30% improvement in response time, even though ITR decreased by 5%. When this occurs, factors such as XA I/O architecture and minidisk caching outweigh the adverse effects of increased processor usage. These factors have a positive effect on response time because they reduce I/O wait time, which is often the largest component of system response time.

Keep in mind that in an actual migration to a new VM release, other factors (such as hardware, licensed product release levels, and workload) are often changed in the same time frame. It is not unusual for the performance effects from upgrading VM to be outweighed by the performance effects from these additional changes.

These VM ITRR estimates can be used in conjunction with the appropriate hardware ITRR figures to estimate the overall performance change that would result from migrating both hardware and VM. For example, suppose that the new processor's ITR is 1.30 times that of the current system and suppose that the migration also includes an upgrade from VM/XA 2.1 to VM/ESA Release 2.1. From the above table, the estimated ITRR for migrating from VM/XA 2.1 to VM/ESA 2.1 is 1.10. Therefore, the estimated overall increase in system capacity is 1.30*1.10 = 1.43.

The above table represents CMS-intensive performance for the case where all files are on minidisks. The release-to-release ITR ratios for shared file system (SFS) usage are very similar to the ones shown here. SFS release-to-release measurement results are provided in the reports listed on page 68.

Derivation and Supporting Data

This section explains how the ITR ratios shown above were derived.

The derivation column in Table 17 on page 70 shows how the stated ITR ratio was calculated. For example, the ITRR of 1.02 for migrating from VM/ESA 1.1 to VM/ESA 2.1 was calculated by multiplying the average ITRR for migrating from VM/ESA 1.1 to VM/ESA 2.0 (R11) by the average ITRR for migrating from VM/ESA 2.0 to VM/ESA 2.1 (R20): 1.01*1.01 = 1.02. R11 was calculated by averaging the ITRRs for VM measurement pairs 15 through 18 (see Table 18 on page 74). Likewise, R20 was calculated by averaging the ITRRs for VM measurement pairs 19 through 21.

Except where noted, any given measurement pair represents two measurements where the only difference is the VM release. As such, all the performance results obtained for one of the measurements in the pair can validly be compared to the corresponding results for the other measurement.

By contrast, there are often substantial environmental differences between unpaired measurements. Factors such as number of users, workload, processor model, and I/O configuration will often be different. This greatly limits the kinds of valid inferences that can be drawn when trying to compare data across two or more measurement pairs. For example, response times are very sensitive to a number of specific environmental factors and therefore should only be compared within a set of controlled, comparable measurements.

For this reason, the above table only covers ITR ratios. Experience has shown that ITR ratios are fairly resistant to changes in the measurement environment. Consequently, combining the ITR ratios observed for individual release transitions (as explained above) provides a reasonably good estimate of the ITR ratio that would result for a migration that spans all those releases.

The ITR ratios shown in Table 17 on page 70 are based on the following pairs of measurements:

Table 18. Derivation and supporting data: VM measurement pairs

Pair Number	Source Run ID	Target Run ID	Processor	Memory	Proc. Util.	Base Pg/cmd	ITR Ratio	Symbol
VM/SP 5 to VM/ESA 1.0 (370 Feature): FS7B0R Workload; Report Number 3								
1	H1SR0091	H17R0090	9221-120	16MB	80	9	0.92	(R5)
VM/SP 6 to VM/ESA 1.0 (370 Feature): FS7B0; Report Number 1								
2	EC4295	EC7603	4381-13	16MB	70	15	1.069	
3	EC4295	EC7603	4381-13	16MB	80	20	1.075	
avg							1.07	(R6)
VM/ESA 1.0 (370 Feature) to VM/ESA 2, 9221-120: FS7B0R; Report Number 3								
4	H17R0090	H15R0091	9221-120	16MB, 32MB	80	11	0.90	(R13a)
VM/ESA 1.0 (370 Feature) to VM/ESA 1.1, 9221-170: FS7B0R; Report Number 2								
5	H17R0281	H14R0287	9221-170	64MB	80	7	0.95	(R13b)
VM/SP HPO 5 to VM/ESA 2: FS7B0R; Report Number 3								
6	L1HR1033	L15R0951	9121-320	64MB/192MB	90	17	0.91	(RHa)
VM/SP HPO 5 to VM/ESA 1.0 (ESA Feature): FS7B0R; Report Number 1								
7	Y25R1141	Y23R1143	3090-200J	64MB/512MB	90	22	0.97	(RHb)
VM/XA 2.0 to VM/XA 2.1: FS7B0R; Report Number 1								
8	Y62R5401	Y6$R5401	3090-600J	512MB/2GB	90	15	1.02	(RX20)
VM/XA 2.1 to VM/ESA 1.0 (ESA Feature): FS7B0R; Report Number 1								
9	Y2$R2001	Y23R2001	3090-200J	256MB/2GB	90	11	1.064	
10	Y6$R5401	Y63R5405	3090-600J	512MB/2GB	90	12	1.029	
avg							1.04	(RX21)
VM/ESA 1.0 (ESA Feature) to VM/ESA 1.1: FS7B0R; Report Number 2								
11	Y63R5866	Y64R5865	9021-720	512MB/2GB	90	13	1.059	
12	L23R1770	L24R1770	9121-480	192MB/64MB	90	13	1.032	
13	L13R0911	L14R0910	9121-320	192MB/64MB	90	12	1.045	
14	H13R0280	H14R0287	9221-170	48M/16MB	80	11	1.043	
avg							1.04	(R1E)
VM/ESA 1.1 to VM/ESA 2: FS7B0R; Report Number 3								
15	264RB424	265RB426	9021-900	1GB/4GB	90	16	1.018	
16	L24R1876	L25R187F	9121-480	192MB/64MB	90	14	1.005	
17	L24R1821	L25R1823	9121-480	128MB/0MB	90	15	1.009	
18	H14R0292	H15R0294	9221-170	48MB/16MB	90	12	1.009	(R11)
avg							1.01	
VM/ESA 2 to VM/ESA 2.1: FS7F0R; Report Number 4								
19	S45E5400	S46E5400	9121-742	1GB/1GB	90	17	1.012	
20	S45E5201	S46E5200	9121-742	320MB/64MB	90	19	1.011	
21	H15E0290	H16E0290	9221-170	48MB/16MB	90	15	1.016	
avg							1.01	(R2)

The report numbers refer to the list of VM performance reports on page 68.

Explanation of columns:

Memory The amount of real storage and (when applicable) expanded storage in the measured configuration.

Proc. Util. Approximate processor utilization. The number of users is adjusted so that the source case runs at or near the stated utilization. The target case is then run with the same number of users.

Base Pg/cmd The average number of paging operations per command measured for the source case. This value gives an indication of how real-memory-constrained the environment is. For configurations with expanded storage used for paging, this value includes expanded storage PGIN and PGOUT operations in addition to DASD page I/Os.

Symbol The symbol used to represent this release transition in Table 17 on page 70.

The FS7B0R workload (CMS-intensive, minidisks, remote users simulated by TPNS) or the (very similar) FS7F0R workload was used for all comparisons except those involving VM/SP 6. For those comparisons, the FS7B0 workload was used (CMS-intensive, minidisks, local users simulated by the full screen internal driver (FSID) tool).

The results in this table illustrate that the release-to-release ITR ratios can and do vary to some extent from one measured environment to another.

Virtual Disk in Storage

The virtual disk in storage function introduced in VM/ESA Release 2.1 can be a valuable technique for improving the performance of VM/ESA and its guest operating systems. This section covers the following topics:

- background information
- degree of benefit
- usage guidelines
- performance monitoring

This is followed by performance measurement results for both the VSE guest and CMS-intensive environments.

Background: A virtual disk in storage is a new type of minidisk and appears to the system like a volatile FBA minidisk that is extremely fast. A virtual disk in storage achieves its high performance by residing in an ESA/370 address space in system storage rather than on DASD. Data on the virtual disk in storage are managed by the VM Control Program (CP) paging subsystem, and may reside in main storage, expanded storage, or paging space on DASD, depending on when the data were last referenced.

Virtual disks in storage can provide better performance than traditional minidisks located on DASD. This improvement is mainly in terms of reduced elapsed time, although ITR can also improve. As with other data-in-memory techniques, the use of virtual disks in storage represents a tradeoff between decreased file I/Os and increased paging I/Os (resulting from increased real storage requirements). Because of this tradeoff, there are some cases where their use will improve overall performance and other cases where overall performance will decrease. The measurement results provided in this section include examples of both cases.

Degree of Benefit: Although the use of virtual disks in storage can reduce processor requirements, the primary performance benefit is the elapsed time reduction that results from eliminating file I/Os. For any given interaction or job, the percentage reduction in elapsed time depends on how many I/Os are eliminated and the average DASD response time of those I/Os. Consider the following example:

- elapsed time: 5 seconds
- file I/Os removed by using a virtual disk in storage: 100
- average DASD response time for those I/Os: 20 milliseconds

In this example, the use of virtual disks in storage would tend to result in an elapsed time decrease of about 2 seconds (100*0.020), which is a 40% reduction. This assumes no significant increase in paging for that interaction. This is often the case for those interactions that use virtual disks in storage. Any increase in paging more typically shows up on a system basis and is distributed among all the users.

In addition to these elapsed time benefits, their use can result in a significant increase in system throughput in cases where it removes or reduces an I/O bottleneck. This can occur, for example, when a virtual disk in storage is used for VSE

DOS.LOCK.FILE. By removing the I/O bottleneck, the system workload can be increased until some other system resource becomes the limiting factor.

Guidelines: Here are some suggestions that will help you to use virtual disks in storage effectively:

1. Heavily used temporary minidisks make good candidates, especially if the amount of referenced data is small.

   The more I/Os there are to a given minidisk, the larger the benefit side of the cost/benefit tradeoff if the files on that minidisk are moved to a virtual disk in storage. If the amount of referenced data is small, the cost side of the cost/benefit tradeoff is small.

2. Virtual disks in storage are especially beneficial if minidisk caching is not available.

   When minidisk caching is in effect for a given minidisk, it may already be eliminating most of the read I/Os, leaving only the write I/Os to be eliminated by moving that minidisk's files to a virtual disk in storage. When minidisk caching is not in effect, both read and write I/Os are eliminated.

3. Consider the alternatives.

   For a given situation, other data-in-memory techniques such as shared segments, SFS DIRCONTROL directories in data spaces, and minidisk caching may be more applicable.

4. Do not use more virtual disks in storage than necessary.

   Each virtual disk in storage has one or more fixed pages associated with it. Being fixed, these pages add to real storage requirements even when the virtual disk in storage is not being used.

   The fixed storage requirement per virtual disk in storage is one page for the segment table (two pages if the virtual disk in storage is larger than one gigabyte) plus one page[3] for each ever-referenced megabyte in its address space.

5. Do not make a virtual disk in storage larger than necessary.

   This is an important consideration if it is to be formatted with a utility that actually writes out every block. This would be the case, for example, when the DSF initialization function is used to format a minidisk to be used by a VSE guest.

   This consideration is less important when the CMS FORMAT command is used to format the virtual disk in storage because CMS FORMAT only references a few of the data pages. However, a small size can be beneficial because of the way CMS allocates blocks. CMS allocates the next available block and does not try to reuse unallocated blocks until it reaches the end of the minidisk and wraps around. With a small size, CMS will reach the end of the minidisk

[3] This page is for the PGMBK, which includes the page table and other information pertaining to that megabyte of virtual storage. It is fixed rather than pageable because the ESA/370 address spaces used to implement virtual disk in storage are considered to be shared system storage. Only private address spaces have pageable page tables.

sooner and therefore start reusing blocks sooner and will confine its references to fewer pages.

6. If necessary, increase the amount of paging space.

   The use of virtual disks in storage will tend to increase the system's paging space requirements. Before making them available on the system, first check to make sure there is enough extra paging space available.

   The DASD_SYSTEM_AREAS VMPRF report can be used for this purpose. It shows the percentage of all page slots that were, on average, in use during the monitored interval. The CP QUERY ALLOC PAGE command (privilege class D) provides similar data on a snapshot basis. It also includes a HIGH PAGE count, which provides additional insight. When HIGH PAGE equals TOTAL PAGES, there was at least one occasion on that page extent when page allocation was forced to wrap around due to end-of-extent before 70% of the preceding page slots become free (the normal criterion). This means that paging space limitations are starting to impair performance.

   After virtual disks in storage have been made available, continue to monitor this information to verify that there is adequate paging space. In addition, you can determine how many page slots are being used for them from the DATA_SPACES_SHARED VMPRF report by summing "DASD Page Slots" over all the virtual disk in storage address spaces.

7. Detach a virtual disk in storage as soon as possible.

   This primarily applies to private virtual disks in storage. When a private virtual disk in storage is detached (or a shared one is detached by the last user), all page frames associated with that virtual disk in storage are made available for other uses. This can greatly reduce the degree to which virtual disk in storage usage increases real storage requirements.

8. Set system and user limits to help prevent excessive use.

   This is done using the CP SET VDISK command. The system limit helps to protect the overall system against overcommitment of storage through excessive use of virtual disks in storage. The user limit, which only applies to virtual disks in storage that are obtained using the CP DEFINE command, prevents any one user from acquiring an excessive amount of space.

   The built-in default for the system limit is intended to provide protection against fixed storage requirements (due to page and segment tables needed to map the address spaces) from using more than 1/4 of the DPA. The built-in default for the user limit is 0, which would restrict usage to virtual disks in storage defined by MDSK directory statements. For more information on the built-in defaults, see the SET VDISK discussion in the *CP Command and Utility Reference*.

9. Monitor performance.

   Monitor overall performance and system paging rate before and after virtual disks in storage are made available. If overall performance shows a decrease that appears to be due to increased paging, you may wish to decrease the system limit for virtual disk in storage usage. If, on the other hand, the system limit is frequently being reached and overall performance is good, you may wish to increase the system limit.

Monitoring Virtual Disk in Storage Performance: The QUERY VDISK command (privilege class B) can be used to display a list of all existing virtual disks in storage. It can also be used to display information about the system or user limits on the amount of virtual storage available for virtual disks in storage. Those limits are set using the SET VDISK command.

The DATA_SPACES_SHARED report is a good source of information about how the system's virtual disks in storage are performing. This report is based on the CP monitor information in domain 3, record 14. Virtual disks in storage show up in this report as entries having an address space name that starts with "VDISK$," followed by the userid of the virtual machine that owns that virtual disk in storage. DATA_SPACES_SHARED provides the following information for each virtual disk in storage that existed at any time during the monitored interval: paging data, resident frames, pages locked, and DASD page slots occupied.

An informal tool called VDISKRPT[4] can be used to obtain the following additional information for each virtual disk in storage: size (in 512-byte blocks), virtual I/Os, virtual I/O rate, and expanded storage pages. VDISKRPT obtains its information from domain 3 record 14 and several other monitor records.

---

[4] Customers can obtain this tool through their IBM representative. It is available on VMTOOLS as VDISKRPT PACKAGE.

VSE Guest

This section examines VSE/ESA 1.2.0 guest performance running under VM/ESA Release 2.1 to illustrate the performance characteristics of virtual disk in storage usage.

The PACE workload was used for all measurements. The PACE workload is batch-only and is characterized by heavy I/O. Three variants, PACEX1, PACEX6 and PACEX8, differ in the volume of work done. See Appendix B, "Workloads" on page 191 for detailed descriptions of the workload.

The configuration for these measurements consisted of:

- dedicated DASD volumes to each guest for the system DASD
- shared VSAM user catalog files defined on shared DASD
- shared DASD containing the VSAM files but not shared files

The VSAM files from one guest were replicated onto the same shared DASD volume for the second guest. Both guests were essentially mirror images. Both VSE guests were run with ESA mode supervisors. The lock file, when placed on shared DASD, was located on the least utilized volume (which contained the user catalog). Several comparisons with the PACE workload using virtual disk in storage were obtained. The PACE workload running with two guests sharing DASD with no virtual disk in storage defined, was used as a base measurement. Then, two other environments were defined:

1. The VSE lock file was located in virtual disk in storage.
2. The VSE lock file and temporary VSAM work areas were located in virtual disks in storage.

Workload: PACEX8

Hardware Configuration

Processor models:	9221-170, 9121-320[2]
Storage	
Real:	128MB, 256MB respectively
Expanded:	0MB, 128MB respectively
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*VSAM*	*VSE Sys.*	*VM Sys.*
3380-A	3880-03	2	1			10	4	4

Virtual Disks in Storage

Type of Virtual Disk	*Size (512 byte blocks)*	*- Number of Virtual Disk in Storage - Lock File*	*VSAM Temporary Files*
FBA	50000	1	
FBA	400000		up to 16

Software Configuration

VSE version: 1.2.0

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
USERVR	1	VSE V=R	24MB/ESA	100		IOASSIST ON CCWTRANS OFF
or						
USERV1	1	VSE V=V	24MB/ESA	100		IOASSIST OFF
SMART	1	RTM	16MB/370	100		
SETUP	1	CP monitor	2MB/XA	100		
USERVF1	1	VSE V=F	24MB/ESA	100		IOASSIST ON
or						
USERV2	1	VSE V=V	24MB/ESA	100		IOASSIST OFF

5 See “Hardware Used” on page 18 for an explanation of how this processor model was defined.

9121-320 / VSE Lock File and Temporary Files

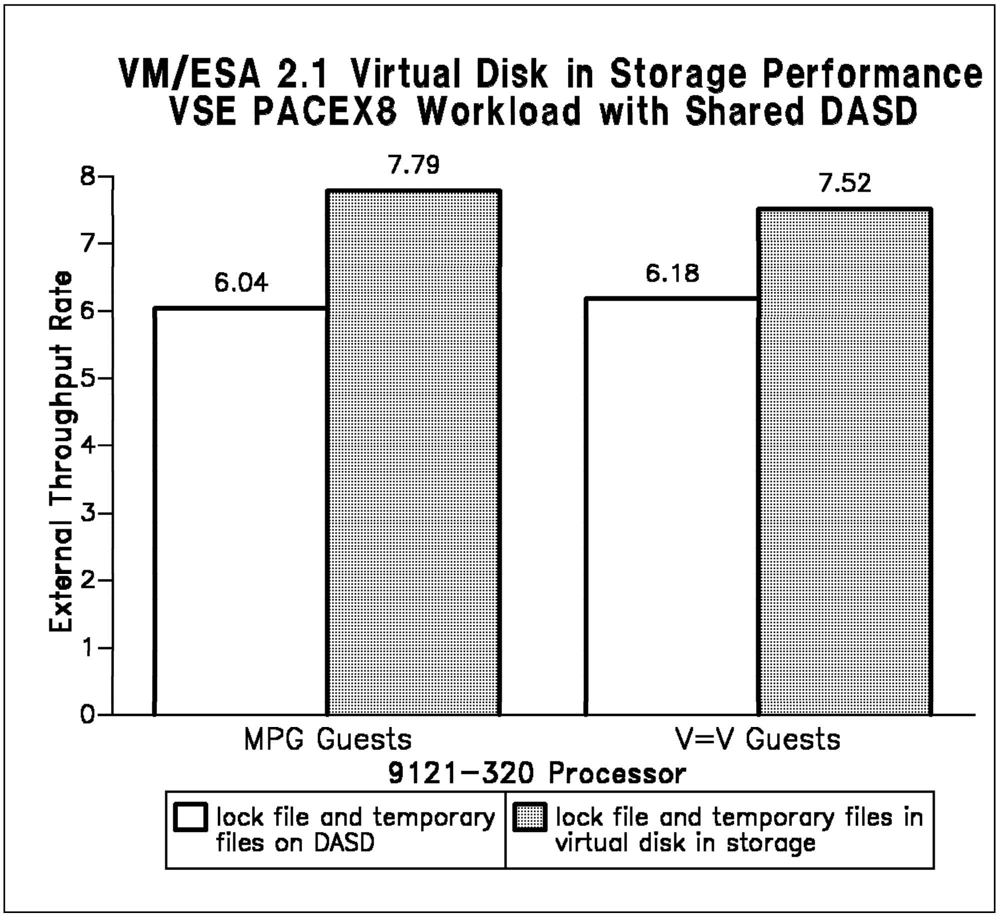

Figure 8. Virtual disk in storage comparison, external throughput rate. PACEX8 workload on VSE under VM/ESA Release 2.1.

Two VSE/ESA guest machines, each running the PACEX8 workload with shared DASD, were run under VM/ESA Release 2.1 on the 9121-320 processor. A multiple preferred guest (MPG) environment (one V=R and one V=F guest) and a V=V environment were measured. The chart compares the guest machines sharing DASD with the VSE lock file and temporary work files located on DASD, versus the VSE lock file and temporary work files on virtual disk in storage. The MPG guest environment had a 29% increase in external throughput rate, and a 2.7% increase in internal throughput rate by placing the lock file and temporary files in virtual disk in storage. The V=V guests had a 22% increase in the external throughput rate, and a 1.9% increase in the internal throughput rate by placing the lock file and temporary files in a virtual disk in storage. By placing the lock file and temporary work files in virtual disk in storage, a 14% reduction in total DASD I/Os (DASD IO TOTAL (V)) was realized. Some paging to and from expanded storage did occur (see XSTOR/CMD) when virtual disk in storage was used. However, the amount of paging was relatively small and the savings in real I/O benefited throughput overall.

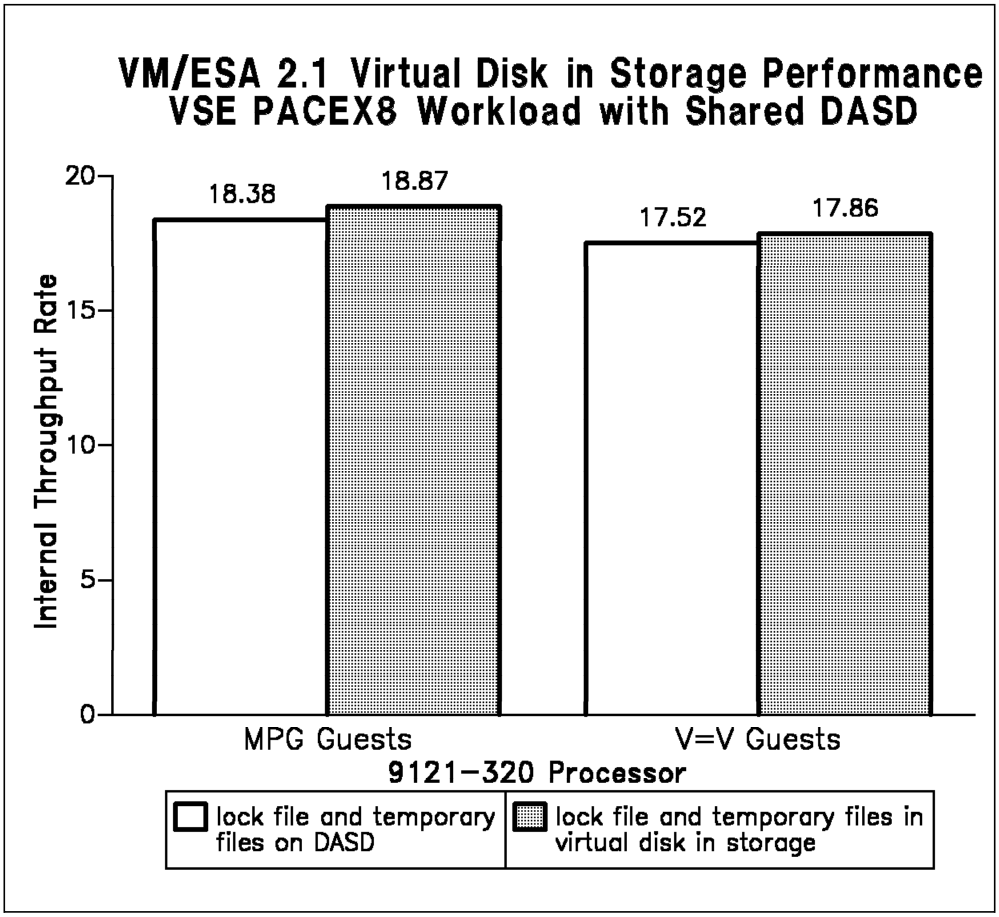

Figure 9. Virtual disk in storage comparison, internal throughput rate. PACEX8 workload on VSE under VM/ESA Release 2.1.

Table 19 (Page 1 of 2). VSE/ESA MPG guests with shared DASD and virtual disk in storage on the 9121-320.				
Virtual Disk in Storage **Run ID** **Release**	**No** **VM/ESA 2.1** **LB68REX9**	**Yes** **VM/ESA 2.1** **LB68RTV3**	**Difference**	**% Difference**
Environment				
Real Storage	**128MB**	**128MB**		
Expanded Storage	**128MB**	**128MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=R,V=F**	**V=R,V=F**		
Number of Guests	**2**	**2**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
Throughput (Min)				
Elapsed Time (H)	1113.0	863.0	-250.0	-22.46%
ETR (H)	6.04	7.79	1.75	28.97%
ITR (H)	18.38	18.87	0.57	2.66%
ITR	18.30	21.63	3.33	18.22%
ITRR (H)	1.000	1.026	0.026	2.65%
ITRR	1.000	1.182	0.182	18.22%
Proc. Usage (Sec)				
PBT/CMD (H)	3.263	3.179	-0.084	-2.58%
PBT/CMD	3.279	2.774	-0.505	-15.41%
CP/CMD (H)	1.188	1.133	-0.055	-4.62%
CP/CMD	1.093	0.925	-0.168	-15.41%
EMUL/CMD (H)	2.075	2.046	-0.029	-1.41%
EMUL/CMD	2.186	1.849	-0.337	-15.41%
Processor Util.				
TOTAL (H)	32.84	41.26	8.42	25.64%
TOTAL	33.00	36.00	3.00	9.09%
TOTAL EMUL (H)	20.88	26.55	5.67	27.15%
TOTAL EMUL	22.00	24.00	2.00	9.09%
TVR(H)	1.57	1.55	-0.02	-1.18%
TVR	1.50	1.50	0.00	0.00%
Storage				
NUCLEUS SIZE (V)	2076KB	2076KB	0	0.00%
TRACE TABLE (V)	64KB	64KB	0	0.00%
PGBLPGS	14386	14179	-207	-1.44%
FREEPGS	91	90	-1	-1.10%
FREE UTIL	0.54	0.54	0.00	0.07%
SHRPGS	299	163	-136	-45.48%
Paging				
PAGE/CMD	0.000	0.000	0.000	na
XSTOR/CMD	0.000	223.455	223.455	na
FAST CLR/CMD	0.000	0.000	0.000	na
I/O				
VIO RATE	304.000	343.000	39.000	12.83%
VIO/CMD	3021.000	2642.937	-378.062	-12.51%
RIO RATE (V)	295.000	316.000	21.000	7.12%
RIO/CMD (V)	2931.562	2434.893	-496.670	-16.94%
DASD IO TOTAL (V)	370068	318923	-51145	-13.82%
DASD IO RATE (V)	324.62	354.36	29.74	9.16%
DASD IO/CMD (V)	3225.92	2730.46	-495.46	-15.36%

Table 19 (Page 2 of 2). VSE/ESA MPG guests with shared DASD and virtual disk in storage on the 9121-320.				
Virtual Disk in Storage **Run ID** **Release**	**No** **VM/ESA 2.1** **LB68REX9**	**Yes** **VM/ESA 2.1** **LB68RTV3**	**Difference**	**% Difference**
Environment				
Real Storage	**128MB**	**128MB**		
Expanded Storage	**128MB**	**128MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=R,V=F**	**V=R,V=F**		
Number of Guests	**2**	**2**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
PRIVOPs				
PRIVOP/CMD	3026.816	2647.862	-378.953	-12.52%
DIAG/CMD	339.058	280.079	-58.979	-17.39%
SIE/CMD	10424.437	9231.018	-1193.420	-11.45%
SIE INTCPT/CMD	9486.238	8215.606	-1270.632	-13.39%
FREE TOTL/CMD	5972.437	3775.625	-2196.812	-36.78%
Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM				

Table 20 (Page 1 of 2). VSE/ESA V=V guests with shared DASD and virtual disk in storage on the 9121-320.				
Virtual Disk in Storage **Release** **Run ID**	**No** **VM/ESA 2.1** **LB68VEV5**	**Yes** **VM/ESA 2.1** **LB68VTV4**	**Difference**	**% Difference**
Environment				
Real Storage	**128MB**	**128MB**		
Expanded Storage	**128MB**	**128MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Number of Guests	**2**	**2**		
Guest Setting	**V=V**	**V=V**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
Throughput (Min)				
Elapsed Time (H)	1088.0	894.0	-194.0	-17.83%
ETR (H)	6.18	7.52	1.34	21.70%
ITR (H)	17.52	17.86	0.34	1.94%
ITR	18.17	17.90	-0.27	-1.48%
ITRR (H)	1.000	1.020	0.020	2.04%
ITRR	1.000	0.985	-0.015	-1.48%
Proc. Usage (Sec)				
PBT/CMD (H)	3.426	3.357	-0.068	-2.00%
PBT/CMD	3.303	3.353	0.050	1.50%
CP/CMD (H)	1.328	1.294	-0.034	-2.57%
CP/CMD	1.166	1.197	0.032	2.71%
EMUL/CMD (H)	2.097	2.063	-0.034	-1.63%
EMUL/CMD	2.137	2.155	0.018	0.84%
Processor Util.				
TOTAL (H)	35.27	42.06	6.80	19.27%
TOTAL	34.00	42.00	8.00	23.53%
TOTAL EMUL (H)	21.59	25.85	4.26	19.71%
TOTAL EMUL	22.00	27.00	5.00	22.73%
TVR(H)	1.63	1.63	-0.01	-0.37%
TVR	1.55	1.56	0.01	0.65%
Storage				
NUCLEUS SIZE (V)	2076KB	2076KB	0	0.00%
TRACE TABLE (V)	64KB	64KB	0	0.00%
PGBLPGS	14373	14164	-209	-1.45%
FREEPGS	91	91	0	0.00%
FREE UTIL	0.56	0.56	-0.01	-0.98%
SHRPGS	279	112	-167	-59.86%
Paging				
PAGE/CMD	0.000	0.000	0.000	na
XSTOR/CMD	0.000	518.839	518.839	na
FAST CLR/CMD	87.429	95.786	8.357	9.56%
I/O				
VIO RATE	333.000	415.000	82.000	24.62%
VIO/CMD	3234.857	3312.589	77.732	2.40%
RIO RATE (V)	341.000	376.000	35.000	10.26%
RIO/CMD (V)	3312.571	3001.286	-311.286	-9.40%
DASD IO TOTAL (V)	367048	315197	-51851	-14.13%
DASD IO RATE (V)	339.86	375.23	35.38	10.41%
DASD IO/CMD (V)	3301.49	2995.18	-306.31	-9.28%

Table 20 (Page 2 of 2). VSE/ESA V=V guests with shared DASD and virtual disk in storage on the 9121-320.

Virtual Disk in Storage **Release** **Run ID**	**No** **VM/ESA 2.1** **LB68VEV5**	**Yes** **VM/ESA 2.1** **LB68VTV4**	**Difference**	**% Difference**
Environment				
Real Storage	**128MB**	**128MB**		
Expanded Storage	**128MB**	**128MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Number of Guests	**2**	**2**		
Guest Setting	**V=V**	**V=V**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
PRIVOPs				
PRIVOP/CMD	3236.887	3315.462	78.575	2.43%
DIAG/CMD	331.035	312.843	-18.192	-5.50%
SIE/CMD	10996.571	11494.286	497.714	4.53%
SIE INTCPT/CMD	9896.914	10114.971	218.057	2.20%
FREE TOTL/CMD	5605.143	4046.946	-1558.196	-27.80%

Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM

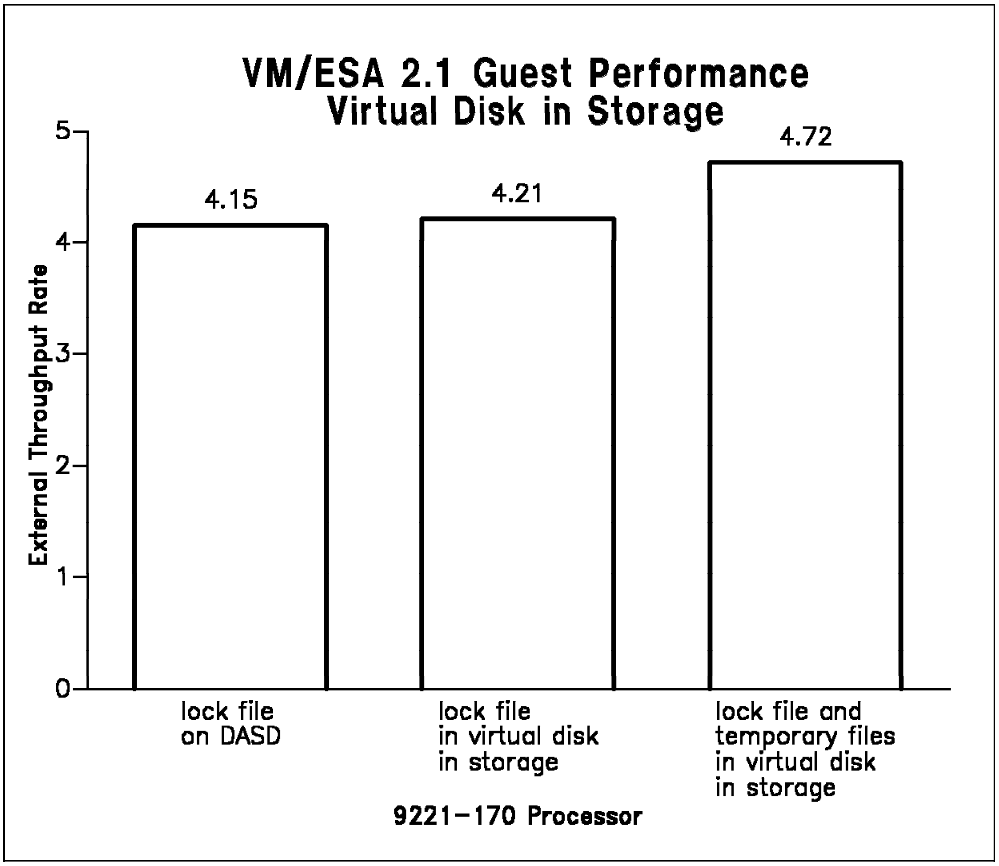

Figure 10. Virtual disk in storage comparisons. PACEX6 workload on two VSE guests under VM/ESA Release 2.1.

Using the MPG facility, two preferred guests, one V=R and one V=F, were run on the 9221-170 processor with shared DASD. Each guest ran 6 copies of the PACE workload in 6 partitions. PACEX6 consists of 6 multiples of the 7 batch jobs that make up the PACE workload. The comparisons in the charts show the external and internal throughput rates for these measurements. Three measurements were obtained:

- 2 preferred guest machines sharing DASD, lock file on shared DASD
- 2 preferred guest machines sharing DASD, with lock file defined in virtual disk in storage
- 2 preferred guest machines sharing DASD, with lock file and temporary work files defined in virtual disk in storage

A virtual disk in storage with 50 000 512-byte blocks of storage was defined for the lock file. Sixteen other virtual disks in storage were also defined for the temporary batch work files with 400 000 512-byte blocks each. For the measurement with the lock file on DASD, 22 584 I/Os occurred to the DASD containing the lock file. The I/O rate to this DASD was about 18.8 I/Os per second and the DASD response time was 15.3 milliseconds. When the lock file was placed in virtual disk in storage, the total I/O count for the same DASD dropped to 4086, at an I/O rate of 3.4 I/Os per second. DASD response time dropped to 11.9 milliseconds. When the lock file was located on a DASD volume, it was not I/O constrained in this measurement environment. Placing the lock file in virtual disk in storage increased the internal throughput by about 2%; the external throughput rate improved by 1%.

Within the 7 jobs that make up one copy of the PACE workload, 3 jobs open a temporary file in VSAM for work space for COBOL compiles, sort activities and SAM work files. These temporary files are prime candidates for placement within a virtual disk in storage.

Normally, when running this workload in a shared DASD configuration with no virtual disks in storage defined, a separate VM paging DASD is not used. However, with the additional virtual disks defined for the temporary batch work files, an additional VM paging DASD was needed.

The internal throughput rate for the measurement with lock file and temporary files in virtual disks in storage gained 4% over the run without virtual disks in storage defined. The external throughput rate increased by 14%.

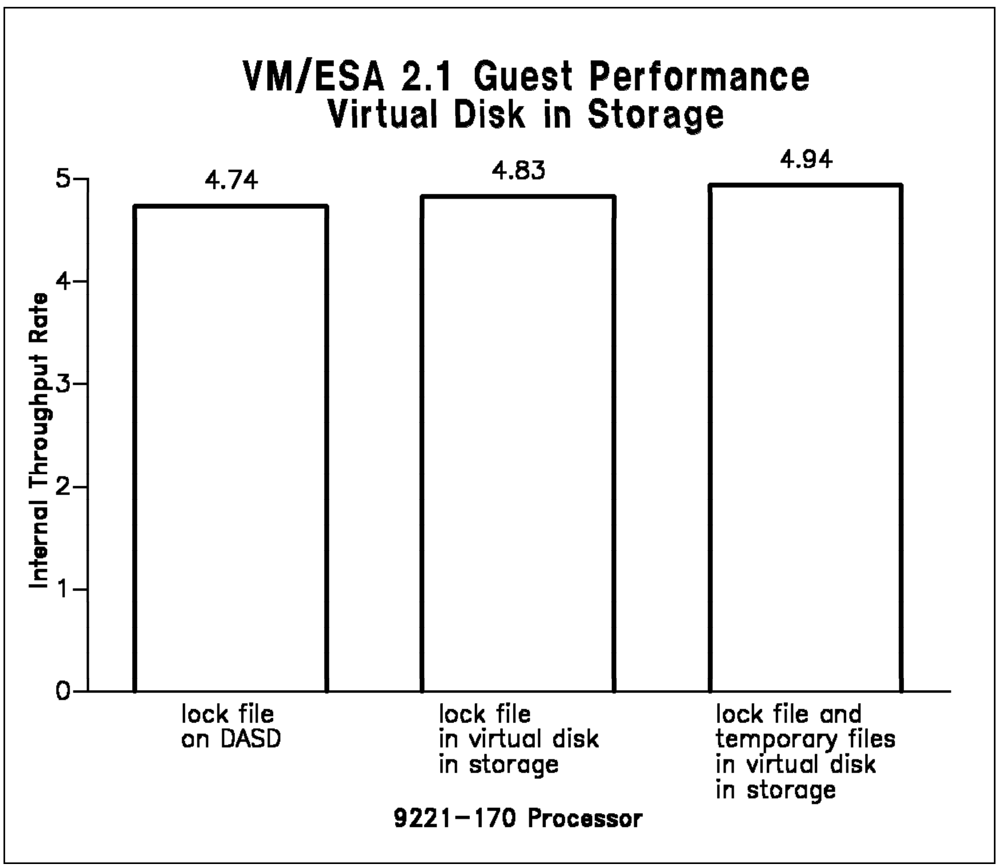

Figure 11. Virtual disk in storage comparisons. PACEX6 workload on two VSE guests under VM/ESA Release 2.1.

Table 21 (Page 1 of 2). VSE/ESA guest with virtual disk in storage used on a 9221-170.

Virtual Disk in Storage	**No**	**Lock file only**	**Lock file and temporary work files**
Release	**VM/ESA 2.1**	**VM/ESA 2.1**	**VM/ESA 2.1**
Run ID	**VR2170S5**	**VR2170S4**	**VR2170S3**
Environment			
IML Mode	**ESA**	**ESA**	**ESA**
Real Storage	**128MB**	**128MB**	**128MB**
Expanded Storage	**0MB**	**0MB**	**0MB**
VM Mode	**ESA**	**ESA**	**ESA**
VM Size	**24MB**	**24MB**	**24MB**
Guest Setting	**V=R,V=F**	**V=R,V=F**	**V=R,V=F**
Number of Guests	**2**	**2**	**2**
VSE Supervisor	**ESA**	**ESA**	**ESA**
Processors	**1**	**1**	**1**
Throughput (Min)			
Elapsed Time (H)	1214.2	1197.7	1068.7
ETR (H)	4.15	4.21	4.72
ITR (H)	4.74	4.83	4.94
ITR	4.72	4.84	4.91
ITRR (H)	1.000	1.019	1.041
ITRR	1.000	1.025	1.041
Proc. Usage (Sec)			
PBT/CMD (H)	12.647	12.410	12.154
PBT/CMD	12.721	12.405	12.214
CP/CMD (H)	6.132	5.889	5.733
CP/CMD	4.915	4.563	4.453
EMUL/CMD (H)	6.515	6.521	6.421
EMUL/CMD	7.806	7.842	7.761
Processor Util.			
TOTAL (H)	87.49	87.03	95.53
TOTAL	88.00	87.00	96.00
TOTAL EMUL (H)	45.07	45.73	50.47
TOTAL EMUL	54.00	55.00	61.00
TVR(H)	1.94	1.90	1.89
TVR	1.63	1.58	1.57
Storage			
NUCLEUS SIZE (V)	2076KB	2076KB	2076KB
TRACE TABLE (V)	64KB	64KB	64KB
PGBLPGS	14368	14368	14419
FREEPGS	89	89	86
FREE UTIL	0.55	0.55	0.56
SHRPGS	436	436	647
Paging			
PAGE/CMD	0.000	0.000	114.506
XSTOR/CMD	0.000	0.000	0.000
FAST CLR/CMD	0.000	0.000	76.337
I/O			
VIO RATE	207.000	218.000	237.000
VIO/CMD	2992.215	3108.357	3015.330
RIO RATE (V)	209.000	194.000	199.000
RIO/CMD (V)	3021.125	2766.152	2531.859
DASD IO TOTAL (V)	276293	258282	239667
DASD IO RATE (V)	230.24	215.24	221.91
DASD IO/CMD (V)	3328.21	3068.93	2823.39

Table 21 (Page 2 of 2). VSE/ESA guest with virtual disk in storage used on a 9221-170.

Virtual Disk in Storage	**No**	**Lock file only**	**Lock file and temporary work files**
Release	**VM/ESA 2.1**	**VM/ESA 2.1**	**VM/ESA 2.1**
Run ID	**VR2170S5**	**VR2170S4**	**VR2170S3**
Environment			
IML Mode	**ESA**	**ESA**	**ESA**
Real Storage	**128MB**	**128MB**	**128MB**
Expanded Storage	**0MB**	**0MB**	**0MB**
VM Mode	**ESA**	**ESA**	**ESA**
VM Size	**24MB**	**24MB**	**24MB**
Guest Setting	**V=R,V=F**	**V=R,V=F**	**V=R,V=F**
Number of Guests	**2**	**2**	**2**
VSE Supervisor	**ESA**	**ESA**	**ESA**
Processors	**1**	**1**	**1**
PRIVOPs			
PRIVOP/CMD	3008.326	3116.077	3020.378
DIAG/CMD	385.331	385.994	359.225
SIE/CMD	10610.075	10907.766	10623.629
SIE INTCPT/CMD	8700.261	9053.445	8817.612
FREE TOTL/CMD	6085.615	4819.379	4440.295

Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM

9221-170 / Single Thread PACEX1

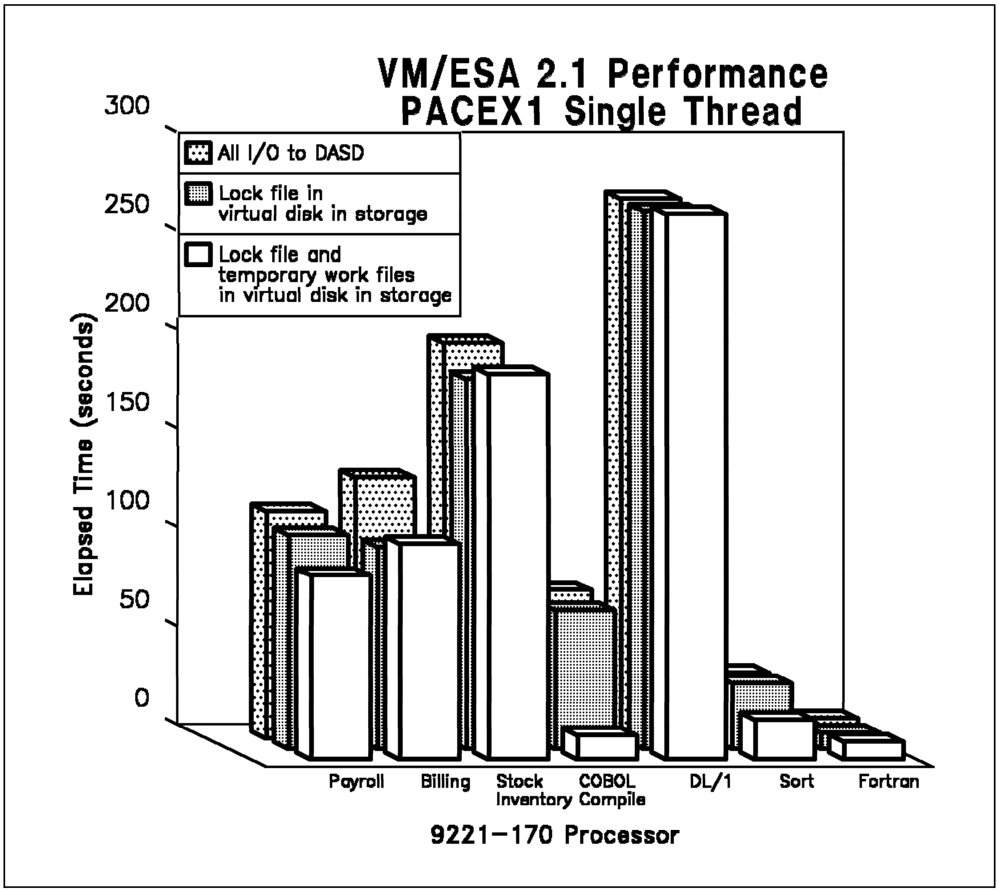

Figure 12. PACEX1 single thread. PACEX1 workload on two preferred VSE guests under VM/ESA Release 2.1.

Using the MPG facility, two VSE/ESA preferred guests were run under VM/ESA Release 2.1 with shared DASD. Both guests ran PACEX1 (7 jobs) in a single partition. In this mode, each job runs in a set order predetermined by the priority assignment to each job. Both guests ran the same job, sharing the same DASD volume. Both guests completed the same batch job at the same time (± 5 sec). Three measurements were obtained:

- lock file defined on a DASD volume
- lock file defined in a virtual disk in storage
- lock file and temporary work files defined in a virtual disk in storage

The tower chart illustrates the elapsed time duration for each job for the three different measurement scenarios. Three of the 7 jobs (Payroll, COBOL, and Sort) use some temporary work files that are ideal for virtual disk in storage use. The COBOL job in particular can use virtual disk in storage for all of the compile and link activities. Elapsed time for the COBOL job went from 75 seconds (all I/O to DASD), to 12 seconds (lock file and temporary work files in virtual disk in storage), an 84% reduction in job run time. The Sort and Payroll jobs have a mixture of I/O to both DASD and virtual disk in storage. Because of this mixture, the Sort and Payroll jobs did not gain as much of a reduction in elapsed time as did the COBOL compile job. The Payroll, Billing, and Stock Inventory applications received a benefit from simply placing the lock file in virtual disk in storage. The lock file was placed on a least-used DASD volume. The benefit of virtual disk in storage use, as illustrated here, depends on how much of a job's elapsed time was due to DASD I/Os that could be eliminated through the use of virtual disk in storage.

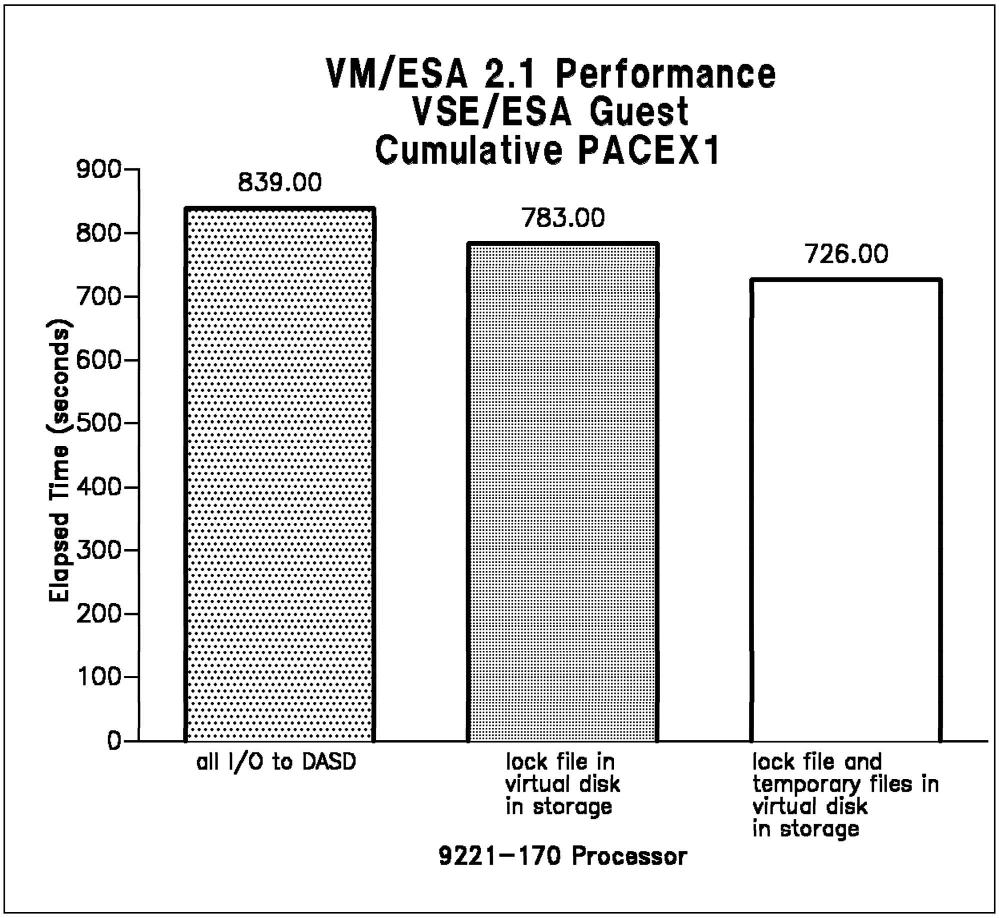

Figure 13. PACEX1 single thread elapsed time. PACEX1 workload on two preferred VSE/ESA guests under VM/ESA Release 2.1.

This chart illustrates the overall reduction in elapsed run time for the PACEX1 workload running on the 9221-170 processor with two preferred guests. By placing the lock file in virtual disk in storage, a 6.6% reduction in elapsed time was obtained. The overall reduction in elapsed time for lock file and temporary files in virtual disk in storage was 13.4%.

Table 22 (Page 1 of 2). VSE/ESA PACEX1 guests of VM/ESA Release 2.1 on the 9221-170.

Virtual Disk in Storage **Release** **Run ID**	**Not used** **VM/ESA 2.1** **VR2170S8**	**Lock file only** **VM/ESA 2.1** **VR2170S7**	**Difference**	**% Difference**
Environment				
IML Mode	**ESA**	**ESA**		
Real Storage	**112MB**	**112MB**		
Exp. Storage	**16MB**	**16MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=R,V=F**	**V=R,V=F**		
Number of Guests	**2**	**2**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
Throughput (Min)				
Elapsed Time (H)	839.1	783.7	-55.3	-6.60%
ETR (H)	1.00	1.07	0.07	7.06%
ITR (H)	4.89	4.97	0.08	1.63%
ITR	4.77	5.36	0.59	12.41%
ITRR (H)	1.000	1.016	0.016	1.63%
ITRR	1.000	1.124	0.124	12.41%
Proc. Usage (Sec)				
PBT/CMD (H)	12.267	12.070	-0.197	-1.60%
PBT/CMD	12.586	11.196	-1.390	-11.04%
CP/CMD (H)	5.929	5.666	-0.263	-4.43%
CP/CMD	5.394	4.478	-0.916	-16.97%
EMUL/CMD (H)	6.338	6.405	0.066	1.04%
EMUL/CMD	7.192	6.718	-0.474	-6.60%
Processor Util.				
TOTAL (H)	20.47	21.56	1.09	5.34%
TOTAL	21.00	20.00	-1.00	-4.76%
TOTAL EMUL (H)	10.58	11.44	0.86	8.18%
TOTAL EMUL	12.00	12.00	0.00	0.00%
TVR(H)	1.94	1.88	-0.05	-2.62%
TVR	1.75	1.67	-0.08	-4.76%
Storage				
NUCLEUS SIZE (V)	2076KB	2076KB	0	0.00%
TRACE TABLE (V)	64KB	64KB	0	0.00%
PGBLPGS	10399	10399	0	0.00%
FREEPGS	85	86	1	1.18%
FREE UTIL	0.57	0.56	-0.01	-1.67%
SHRPGS	733	733	0	0.00%
Paging				
PAGE/CMD	0.000	0.000	0.000	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	0.000	0.000	0.000	na
I/O				
VIO RATE	50.000	52.000	2.000	4.00%
VIO/CMD	2996.681	2911.010	-85.671	-2.86%
RIO RATE (V)	51.000	50.000	-1.000	-1.96%
RIO/CMD (V)	3056.615	2799.048	-257.567	-8.43%
DASD IO TOTAL (V)	46595	43022	-3573	-7.67%
DASD IO RATE (V)	55.47	55.16	-0.31	-0.57%
DASD IO/CMD (V)	3324.53	3087.71	-236.82	-7.12%

Table 22 (Page 2 of 2). VSE/ESA PACEX1 guests of VM/ESA Release 2.1 on the 9221-170.				
Virtual Disk in Storage **Release** **Run ID**	**Not used** **VM/ESA 2.1** **VR2170S8**	**Lock file only** **VM/ESA 2.1** **VR2170S7**	**Difference**	**% Difference**
Environment				
IML Mode	**ESA**	**ESA**		
Real Storage	**112MB**	**112MB**		
Exp. Storage	**16MB**	**16MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=R,V=F**	**V=R,V=F**		
Number of Guests	**2**	**2**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
PRIVOPs				
PRIVOP/CMD	3049.741	2935.381	-114.359	-3.75%
DIAG/CMD	1011.067	940.653	-70.414	-6.96%
SIE/CMD	10907.919	10468.439	-439.480	-4.03%
SIE INTCPT/CMD	10580.682	10154.386	-426.296	-4.03%
FREE TOTL/CMD	7971.172	6493.791	-1477.381	-18.53%

Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM

Table 23 (Page 1 of 2). VSE/ESA PACEX1 guests of VM/ESA Release 2.1 on the 9221-170.

Virtual Disk in Storage Release Run ID	Not used VM/ESA 2.1 VR2170S8	Lock file and temporary work files VM/ESA 2.1 VR2170S6	Difference	% Difference
Environment				
IML Mode	**ESA**	**ESA**		
Real Storage	**112MB**	**112MB**		
Exp. Storage	**16MB**	**16MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=R,V=F**	**V=R,V=F**		
Number of Guests	**2**	**2**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
Throughput (Min)				
Elapsed Time (H)	839.1	726.6	-112.4	-13.40%
ETR (H)	1.00	1.16	0.15	15.48%
ITR (H)	4.89	5.08	0.19	3.83%
ITR	4.77	5.50	0.74	15.48%
ITRR (H)	1.000	1.038	0.038	3.83%
ITRR	1.000	1.155	0.155	15.48%
Proc. Usage (Sec)				
PBT/CMD (H)	12.267	11.815	-0.452	-3.68%
PBT/CMD	12.586	10.899	-1.687	-13.40%
CP/CMD (H)	5.929	5.516	-0.412	-6.96%
CP/CMD	5.394	4.671	-0.723	-13.40%
EMUL/CMD (H)	6.338	6.299	-0.040	-0.62%
EMUL/CMD	7.192	6.228	-0.964	-13.40%
Processor Util.				
TOTAL (H)	20.47	22.76	2.30	11.22%
TOTAL	21.00	21.00	0.00	0.00%
TOTAL EMUL (H)	10.58	12.14	1.56	14.75%
TOTAL EMUL	12.00	12.00	0.00	0.00%
TVR(H)	1.94	1.88	-0.06	-3.08%
TVR	1.75	1.75	0.00	0.00%
Storage				
NUCLEUS SIZE (V)	2076KB	2076KB	0	0.00%
TRACE TABLE (V)	64KB	64KB	0	0.00%
PGBLPGS	10399	10399	0	0.00%
FREEPGS	85	86	1	1.18%
FREE UTIL	0.57	0.56	-0.01	-1.67%
SHRPGS	733	702	-31	-4.23%
Paging				
PAGE/CMD	0.000	0.000	0.000	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	0.000	0.000	0.000	na
I/O				
VIO RATE	50.000	54.000	4.000	8.00%
VIO/CMD	2996.681	2802.687	-193.994	-6.47%
RIO RATE (V)	51.000	50.000	-1.000	-1.96%
RIO/CMD (V)	3056.615	2595.080	-461.534	-15.10%
DASD IO TOTAL (V)	46595	39633	-6962	-14.94%
DASD IO RATE (V)	55.47	55.05	-0.42	-0.77%
DASD IO/CMD (V)	3324.53	2856.97	-467.57	-14.06%

Table 23 (Page 2 of 2). VSE/ESA PACEX1 guests of VM/ESA Release 2.1 on the 9221-170.

Virtual Disk in Storage Release Run ID	Not used VM/ESA 2.1 VR2170S8	Lock file and temporary work files VM/ESA 2.1 VR2170S6	Difference	% Difference
Environment				
IML Mode	**ESA**	**ESA**		
Real Storage	**112MB**	**112MB**		
Exp. Storage	**16MB**	**16MB**		
VM Mode	**ESA**	**ESA**		
VM Size	**24MB**	**24MB**		
Guest Setting	**V=R,V=F**	**V=R,V=F**		
Number of Guests	**2**	**2**		
VSE Supervisor	**ESA**	**ESA**		
Processors	**1**	**1**		
PRIVOPs				
PRIVOP/CMD	3049.741	2809.884	-239.856	-7.86%
DIAG/CMD	1011.067	881.271	-129.796	-12.84%
SIE/CMD	10907.919	10017.010	-890.909	-8.17%
SIE INTCPT/CMD	10580.682	9716.500	-864.182	-8.17%
FREE TOTL/CMD	7971.172	5968.685	-2002.487	-25.12%

Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM

CMS Environment

There are a number of ways in which virtual disk in storage might be used to advantage in the CMS environment. One way is as a replacement for CMS temporary minidisks (T-disks). The performance measurements presented in this section explore this type of usage.

Another type of use takes advantage of the fact that virtual disks in storage can be shared among users. In many cases, existing functions such as minidisk caching, SFS DIRCONTROL directories using VM Data Spaces, and shared segments are quite adequate methods for reducing file I/Os by keeping frequently referenced data in memory. However, there are situations where these methods do not apply but virtual disks in storage can be used instead. For example, systems that do not have expanded storage cannot use minidisk caching.

Relative to T-disks, virtual disks in storage offer significant performance advantages such as near-instantaneous formatting time and the ability to capitalize on available memory to reduce physical I/Os. However, as the performance results in this section illustrate, inappropriate use of virtual disks in storage as a T-disk replacement can potentially reduce overall system performance as a result of increased real storage requirements.

Two sets of performance measurements are provided. The first set represents the case where virtual disks in storage are substituted for T-disks that are held for a relatively short period of time and are then detached when they are no longer needed. The second set represents the case where they are substituted for T-disks that are acquired when the user logs on and are held until the user logs off, even though they are only used occasionally.

Normal use of T-disks is somewhere between these two extremes. The first case (detach after use) is a more favorable case for virtual disk in storage usage because detaching a virtual disk in storage serves as a signal to CP that its page frames are no longer needed and can therefore be immediately made available to the rest of the system for other uses.

Virtual disk in storage performance is also affected by real storage availability. The more real memory the system has, the better virtual disks in storage will perform relative to T-disks. The measurement results in this section illustrate this.

Virtual disk in storage performance is influenced by the CMS block size that is selected when the virtual disk in storage is formatted. For the measured cases, 1KB (the default) provided better performance than 4KB because it resulted in a smaller increase in real storage requirements.

CMS Temporary Minidisks (Detach After Use)

In the FS7F CMS-intensive workload, one of the 17 workload scripts (COB417) acquires a 2-cylinder and a 3-cylinder temporary minidisk, uses them during the execution of a COBOL program, and then detaches them. About 10% of all minidisk I/Os in the workload are directed to these T-disks. Because the T-disks are held during most of the time that the COB417 script is active and because COB417 is executed 5% of the time, each user has these two T-disks attached about 5% of the time. See Appendix B, “Workloads” on page 191 for a description of the FS7F workload.

In this section, measurements made with virtual disks in storage substituted for the two T-disks are compared to standard FS7F measurements. This is intended to represent the case where virtual disks in storage are substituted for temporary minidisks that are detached as soon as they are no longer being used. Two pairs of comparison measurements were obtained on a 9121-480 processor:

- 256MB total storage (nonconstrained)
- 128MB total storage (storage constrained)

This is followed by a comparison of 1KB-formatted virtual disks in storage to 4KB-formatted virtual disks in storage.

9121-480 / 256MB

Workload: FS7F0R

Hardware Configuration

Processor model:	9121-480
Processors used:	2
Storage	
Real:	224MB
Expanded:	32MB (all for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-3	2						2 R
3390-2	3990-2	4	16	5	6			
3390-2	3990-3	4		2	2	16 R		

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB for minidisks
	1KB for virtual disks in storage

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
SMART	1	RTM	16MB/370	3%	400	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	512	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	1860	User	3MB/XC	100		

Measurement Discussion: Overall system performance improved somewhat when virtual disks in storage were substituted for the temporary minidisks. External response time (AVG LAST (T)) decreased 6.0% and the internal throughput rate (ITR (H)) increased 0.3%.

The use of virtual disks in storage represents a trade-off between decreased minidisk I/Os and increased page I/Os (resulting from increased real storage requirements). Because they were detached immediately after use, the increase in real storage requirements was kept to a minimum. In addition, the measured system had ample real storage and therefore was able to handle the additional real storage requirements with only a small increase in paging. Paging (PAGE IO/CMD (V)) increased by only 0.01 page I/Os per command, while minidisk I/Os decreased significantly, as evidenced by the 0.43 decrease in non-paging I/Os per command (NONPAGE RIO/CMD (V)).

Most of the improvement in overall response time came from large response time reductions experienced by the commands that had been using the T-disks. The time required to do the FORMAT went from 2.48 seconds to 0.20 seconds—a 92% decrease. The time required to execute a COBOL program whose work files were on temporary disks went from 6.72 seconds to 2.43 seconds—a 64% decrease. There was little change to the response time of most of the other commands in the workload.

The 0.20 seconds observed as external response time for formatting the virtual disk in storage mainly consists of network communication time. The time required to define, format, and (later) detach a virtual disk in storage is very short and little processing is required. Because of this, it is much more practical to repeatedly acquire and detach virtual disks in storage as needed than it is for T-disks.

In the measured environment, the T-disk volumes were eligible for, and participated in minidisk caching (MDC). Minidisk caching is an alternative way to reduce real DASD I/Os to CMS files. However, minidisk caching assumes that the data is persistent (non-volatile) and therefore does store-through I/O to DASD on write operations. If the files do not need to be persistent, the use of virtual disks in storage has a performance advantage relative to MDC in that even write I/Os can be eliminated (if sufficient real storage is available). In the comparison shown here, this is evidenced by the 21% decrease in MDC WRITES and the 25% decrease in MDC MODS.

Table 24 (Page 1 of 3). Use of virtual disks in storage as short term CMS T-disks / 9121-480 with 256MB of storage

Virtual Disk in Storage **Release** **Run ID**	**Not used** **VM/ESA 2.1** **L26E1868**	**T-disks** **VM/ESA 2.1** **L26E1867**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.123	0.117	-0.006	-4.88%
NONTRIV INT	0.453	0.419	-0.034	-7.51%
TOT INT	0.344	0.319	-0.025	-7.27%
TOT INT ADJ	0.303	0.282	-0.022	-7.22%
AVG FIRST (T)	0.247	0.247	0.000	-0.20%
AVG LAST (T)	0.378	0.355	-0.022	-5.95%
Throughput				
AVG THINK (T)	26.29	26.32	0.03	0.10%
ETR	57.57	57.67	0.10	0.17%
ETR (T)	65.25	65.33	0.08	0.12%
ETR RATIO	0.882	0.883	0.000	0.06%
ITR (H)	73.02	73.21	0.19	0.27%
ITR	32.22	32.33	0.12	0.36%
EMUL ITR	47.55	47.74	0.19	0.40%
ITRR (H)	1.000	1.003	0.003	0.27%
ITRR	1.000	1.004	0.004	0.36%
Proc. Usage				
PBT/CMD (H)	27.391	27.318	-0.073	-0.26%
PBT/CMD	27.431	27.246	-0.185	-0.68%
CP/CMD (H)	9.416	9.397	-0.018	-0.20%
CP/CMD	8.888	8.725	-0.163	-1.84%
EMUL/CMD (H)	17.975	17.921	-0.054	-0.30%
EMUL/CMD	18.543	18.521	-0.022	-0.12%
Processor Util.				
TOTAL (H)	178.74	178.47	-0.26	-0.15%
TOTAL	179.00	178.00	-1.00	-0.56%
UTIL/PROC (H)	89.37	89.24	-0.13	-0.15%
UTIL/PROC	89.50	89.00	-0.50	-0.56%
TOTAL EMUL (H)	117.29	117.08	-0.22	-0.18%
TOTAL EMUL	121.00	121.00	0.00	0.00%
MASTER TOTAL (H)	89.04	88.94	-0.10	-0.11%
MASTER TOTAL	89.00	89.00	0.00	0.00%
MASTER EMUL (H)	52.67	52.54	-0.13	-0.25%
MASTER EMUL	54.00	54.00	0.00	0.00%
TVR(H)	1.52	1.52	0.00	0.04%
TVR	1.48	1.47	-0.01	-0.56%

Table 24 (Page 2 of 3). Use of virtual disks in storage as short term CMS T-disks / 9121-480 with 256MB of storage

Virtual Disk in Storage Release Run ID	Not used VM/ESA 2.1 L26E1868	T-disks VM/ESA 2.1 L26E1867	Difference	% Difference
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Storage				
NUCLEUS SIZE (V)	2364KB	2364KB	0KB	0.00%
TRACE TABLE (V)	400KB	400KB	0KB	0.00%
WKSET (V)	82	81	-1	-1.22%
PGBLPGS	47770	47541	-229	-0.48%
PGBLPGS/USER	25.7	25.6	-0.1	-0.48%
FREEPGS	4953	4970	17	0.34%
FREE UTIL	0.93	0.93	0.00	-0.34%
SHRPGS	1134	1098	-36	-3.17%
Paging				
READS/SEC	625	629	4	0.64%
WRITES/SEC	422	425	3	0.71%
PAGE/CMD	16.045	16.133	0.088	0.55%
PAGE IO RATE (V)	170.200	170.800	0.600	0.35%
PAGE IO/CMD (V)	2.608	2.614	0.006	0.23%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.467	6.964	0.498	7.69%
Queues				
DISPATCH LIST	37.02	36.16	-0.86	-2.32%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	627	617	-10	-1.59%
VIO/CMD	9.608	9.444	-0.164	-1.71%
RIO RATE (V)	391	364	-27	-6.91%
RIO/CMD (V)	5.992	5.572	-0.420	-7.01%
NONPAGE RIO/CMD (V)	3.384	2.958	-0.426	-12.59%
MDC READS	361	361	0	0.00%
MDC WRITES	179	142	-37	-20.67%
MDC MODS	145	109	-36	-24.83%
MDC HIT RATIO	0.91	0.91	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	17.584	17.566	-0.019	-0.11%
DIAG/CMD	26.983	26.951	-0.032	-0.12%
DIAG 08/CMD	0.760	0.759	-0.001	-0.18%
DIAG 14/CMD	0.025	0.024	0.000	-0.88%
DIAG 58/CMD	1.251	1.251	0.000	0.00%
DIAG 98/CMD	1.227	1.231	0.003	0.28%
DIAG A4/CMD	3.913	3.858	-0.055	-1.40%
DIAG A8/CMD	1.888	1.792	-0.096	-5.07%
DIAG 214/CMD	12.941	12.926	-0.015	-0.12%
SIE/CMD	54.586	54.813	0.227	0.42%
SIE INTCPT/CMD	37.664	37.821	0.156	0.42%
FREE TOTL/CMD	64.470	63.920	-0.550	-0.85%

Table 24 (Page 3 of 3). Use of virtual disks in storage as short term CMS T-disks / 9121-480 with 256MB of storage

Virtual Disk in Storage Release Run ID	Not used VM/ESA 2.1 L26E1868	T-disks VM/ESA 2.1 L26E1867	Difference	% Difference
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
VTAM Machines				
WKSET (V)	515	516	1	0.19%
TOT CPU/CMD (V)	3.8737	3.8777	0.0040	0.10%
CP CPU/CMD (V)	1.6431	1.6497	0.0066	0.40%
VIRT CPU/CMD (V)	2.2306	2.2280	-0.0026	-0.12%
DIAG 98/CMD (V)	1.227	1.231	0.003	0.27%

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

9121-480 / 128MB

Workload: FS7F0R

Hardware Configuration

Processor model:	9121-480
Processors used:	2
Storage	
Real:	96MB
Expanded:	32MB (all for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-3	2						2 R
3390-2	3990-2	4	16	5	6			
3390-2	3990-3	4		2	2	16 R		

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB for minidisks
	1KB for virtual disks in storage

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
SMART	1	RTM	16MB/370	3%	400	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	480	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	1820	User	3MB/XC	100		

Measurement Discussion: This 128MB configuration is much more storage constrained than the 256MB comparison described on page 100. As a result, the increased real storage usage, caused by the use of virtual disks in storage, resulted in a much larger increase in paging such that overall performance decreased somewhat when going from T-disks to virtual disks in storage. External response time (AVG LAST (T)) increased by 3% (within measurement variability). Processor capacity (ITR (H)) decreased by 0.4%, mostly as a result of increased processor usage by CP to support the additional paging.

Non-paging I/Os per command (NONPAGE RIO/CMD (V)) decreased by 0.42, very close to the amount of improvement observed for the 256MB comparison. However, page I/Os per command (PAGE IO/CMD (V)) increased by 0.31, much more than the 0.01 increase observed for the 256MB comparison. In this storage-constrained environment, the benefits and cost of virtual disk in storage usage roughly balanced each other.

Even though this is a storage-constrained environment, the commands that actually used the virtual disks in storage still experienced large external response time reductions. FORMAT time decreased 88% and the COBOL program whose work files were on temporary disks had a 50% response time reduction. These improvements are nearly as large as those seen for the 256MB comparison. These results, combined with the fact that external response time increased slightly, mean that those commands in the workload that do not use virtual disks in storage experienced an average increase in response time. This is a side effect of the overall increase in paging caused by their use.

Table 25 (Page 1 of 3). Use of virtual disks in storage as short term CMS T-disks / 9121-480 with 128MB of storage

Virtual Disk in Storage **Release** **Run ID**	**Not used** **VM/ESA 2.1** **L26E1825**	**T-disks** **VM/ESA 2.1** **L26E1828**	**Difference**	**% Difference**
Environment				
Real Storage	**96MB**	**96MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1820**	**1820**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.176	0.172	-0.004	-2.27%
NONTRIV INT	0.600	0.596	-0.004	-0.67%
TOT INT	0.454	0.446	-0.008	-1.76%
TOT INT ADJ	0.410	0.409	-0.001	-0.36%
AVG FIRST (T)	0.342	0.365	0.024	6.87%
AVG LAST (T)	0.520	0.535	0.016	2.98%
Throughput				
AVG THINK (T)	26.34	26.34	-0.01	-0.04%
ETR	57.43	58.28	0.85	1.48%
ETR (T)	63.53	63.56	0.03	0.05%
ETR RATIO	0.904	0.917	0.013	1.43%
ITR (H)	69.51	69.21	-0.30	-0.43%
ITR	31.43	31.75	0.33	1.05%
EMUL ITR	47.80	48.52	0.71	1.49%
ITRR (H)	1.000	0.996	-0.004	-0.43%
ITRR	1.000	1.010	0.010	1.05%
Proc. Usage				
PBT/CMD (H)	28.772	28.897	0.125	0.43%
PBT/CMD	28.807	28.950	0.143	0.50%
CP/CMD (H)	10.617	10.749	0.131	1.24%
CP/CMD	9.917	10.070	0.153	1.54%
EMUL/CMD (H)	18.154	18.148	-0.007	-0.04%
EMUL/CMD	18.890	18.881	-0.009	-0.05%

Table 25 (Page 2 of 3). Use of virtual disks in storage as short term CMS T-disks / 9121-480 with 128MB of storage

Virtual Disk in Storage Release Run ID	Not used VM/ESA 2.1 L26E1825	T-disks VM/ESA 2.1 L26E1828	Difference	% Difference
Environment				
Real Storage	**96MB**	**96MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1820**	**1820**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Processor Util.				
TOTAL (H)	182.78	183.66	0.88	0.48%
TOTAL	183.00	184.00	1.00	0.55%
UTIL/PROC (H)	91.39	91.83	0.44	0.48%
UTIL/PROC	91.50	92.00	0.50	0.55%
TOTAL EMUL (H)	115.33	115.34	0.01	0.01%
TOTAL EMUL	120.00	120.00	0.00	0.00%
MASTER TOTAL (H)	91.13	91.49	0.36	0.40%
MASTER TOTAL	91.00	91.00	0.00	0.00%
MASTER EMUL (H)	51.40	51.20	-0.20	-0.38%
MASTER EMUL	54.00	54.00	0.00	0.00%
TVR(H)	1.58	1.59	0.01	0.47%
TVR	1.53	1.53	0.01	0.55%
Storage				
NUCLEUS SIZE (V)	2352KB	2352KB	0KB	0.00%
TRACE TABLE (V)	400KB	400KB	0KB	0.00%
WKSET (V)	63	61	-2	-3.17%
PGBLPGS	15420	15185	-235	-1.52%
PGBLPGS/USER	8.5	8.3	-0.1	-1.52%
FREEPGS	4762	4779	17	0.36%
FREE UTIL	0.97	0.96	0.00	-0.36%
SHRPGS	1018	1051	33	3.24%
Paging				
READS/SEC	1014	1062	48	4.73%
WRITES/SEC	753	821	68	9.03%
PAGE/CMD	27.815	29.627	1.812	6.51%
PAGE IO RATE (V)	329.000	348.800	19.800	6.02%
PAGE IO/CMD (V)	5.179	5.488	0.309	5.97%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.800	7.316	0.516	7.59%
Queues				
DISPATCH LIST	45.91	47.70	1.79	3.89%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	596	583	-13	-2.18%
VIO/CMD	9.382	9.173	-0.209	-2.23%
RIO RATE (V)	528	521	-7	-1.33%
RIO/CMD (V)	8.312	8.197	-0.114	-1.37%
NONPAGE RIO/CMD (V)	3.133	2.709	-0.424	-13.53%
MDC READS	352	351	-1	-0.28%
MDC WRITES	174	139	-35	-20.11%
MDC MODS	141	106	-35	-24.82%
MDC HIT RATIO	0.91	0.91	0.00	0.00%

Table 25 (Page 3 of 3). Use of virtual disks in storage as short term CMS T-disks / 9121-480 with 128MB of storage

Virtual Disk in Storage Release Run ID	Not used VM/ESA 2.1 L26E1825	T-disks VM/ESA 2.1 L26E1828	Difference	% Difference
Environment				
Real Storage	**96MB**	**96MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1820**	**1820**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
PRIVOPs				
PRIVOP/CMD	17.587	17.669	0.082	0.46%
DIAG/CMD	26.701	26.633	-0.068	-0.25%
DIAG 08/CMD	0.759	0.762	0.004	0.47%
DIAG 14/CMD	0.024	0.025	0.000	1.53%
DIAG 58/CMD	1.250	1.251	0.001	0.10%
DIAG 98/CMD	0.987	0.969	-0.018	-1.80%
DIAG A4/CMD	3.913	3.856	-0.056	-1.44%
DIAG A8/CMD	1.910	1.779	-0.131	-6.86%
DIAG 214/CMD	12.856	12.849	-0.006	-0.05%
SIE/CMD	58.133	58.703	0.570	0.98%
SIE INTCPT/CMD	37.205	36.983	-0.222	-0.60%
FREE TOTL/CMD	64.147	63.502	-0.644	-1.00%
VTAM Machines				
WKSET (V)	482	481	-1	-0.21%
TOT CPU/CMD (V)	3.8567	3.8461	-0.0106	-0.27%
CP CPU/CMD (V)	1.6441	1.6521	0.0080	0.49%
VIRT CPU/CMD (V)	2.2126	2.1940	-0.0186	-0.84%
DIAG 98/CMD (V)	0.987	0.969	-0.017	-1.75%

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

9121-480 / 128MB / 1KB vs. 4KB Block Size

Measurement Configuration: See page 104.

Measurement Discussion: The default CMS block size is 1KB for FBA devices and 4KB for CKD devices. Virtual disk in storage emulates an FBA device, so the default block size for a CMS-formatted virtual disk in storage is 1KB. For actual DASD, 4KB is usually a better choice than 1KB. Because of this, the question arises as to whether this is also true for virtual disks in storage. To help answer that question, a pair of measurements was obtained on the 128MB configuration to compare the performance of 1KB and 4KB-formatted virtual disks in storage.

The results showed that the 4KB-formatted virtual disks in storage did not perform as well as the 1KB-formatted ones. External response time (AVG LAST (T)) increased by 23% (0.122 seconds) and the internal throughput rate (ITR (H)) decreased by 0.4%.

The 4KB block size has the advantage of causing fewer minidisk I/Os but has the disadvantage of using more real storage. The results show that, in this storage-constrained environment, this disadvantage prevailed. As evidenced by NONPAGE RIO/CMD, minidisk I/Os decreased by 0.038 I/Os per command. However, page I/Os per command (PAGE IO/CMD (V)) increased by 0.290.

The decrease in minidisk I/Os is so small because of CMS′s read-ahead, write-behind function. This function chains together up to 8KB (default) blocks at a time in a single I/O request. This is true regardless of what the CMS block size is. Typically, most of the minidisk I/Os done by CMS either make use of this function or have the chaining of blocks controlled by the application or involve just a single block. The combination of these factors eliminates most of the I/O reduction that would otherwise occur when changing from a 1KB to a 4KB block size.

Table 26 (Page 1 of 2). Use of virtual disks in storage as short term CMS T-disks / 9121-480 with 128MB of storage / 1KB vs. 4KB block size

Block Size **Release** **Run ID**	**1KB** **VM/ESA 2.1** **L26E1828**	**4KB** **VM/ESA 2.1** **L26E1826**	**Difference**	**% Difference**
Environment				
Real Storage	**96MB**	**96MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1820**	**1820**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.172	0.172	0.000	0.00%
NONTRIV INT	0.596	0.646	0.050	8.39%
TOT INT	0.446	0.470	0.024	5.38%
TOT INT ADJ	0.409	0.453	0.044	10.84%
AVG FIRST (T)	0.365	0.453	0.088	24.08%
AVG LAST (T)	0.535	0.657	0.122	22.69%
Throughput				
AVG THINK (T)	26.34	26.41	0.07	0.27%
ETR	58.28	61.07	2.79	4.79%
ETR (T)	63.56	63.32	-0.23	-0.37%
ETR RATIO	0.917	0.964	0.047	5.18%
ITR (H)	69.21	68.92	-0.30	-0.43%
ITR	31.75	33.24	1.49	4.69%
EMUL ITR	48.52	51.11	2.59	5.34%
ITRR (H)	1.000	0.996	-0.004	-0.43%
ITRR	1.000	1.047	0.047	4.69%
Proc. Usage				
PBT/CMD (H)	28.897	29.021	0.124	0.43%
PBT/CMD	28.950	29.058	0.107	0.37%
CP/CMD (H)	10.749	10.908	0.159	1.48%
CP/CMD	10.070	10.107	0.037	0.37%
EMUL/CMD (H)	18.148	18.113	-0.035	-0.19%
EMUL/CMD	18.881	18.951	0.070	0.37%
Processor Util.				
TOTAL (H)	183.66	183.76	0.11	0.06%
TOTAL	184.00	184.00	0.00	0.00%
UTIL/PROC (H)	91.83	91.88	0.05	0.06%
UTIL/PROC	92.00	92.00	0.00	0.00%
TOTAL EMUL (H)	115.34	114.69	-0.65	-0.56%
TOTAL EMUL	120.00	120.00	0.00	0.00%
MASTER TOTAL (H)	91.49	91.57	0.07	0.08%
MASTER TOTAL	91.00	92.00	1.00	1.10%
MASTER EMUL (H)	51.20	50.89	-0.31	-0.60%
MASTER EMUL	54.00	53.00	-1.00	-1.85%
TVR(H)	1.59	1.60	0.01	0.62%
TVR	1.53	1.53	0.00	0.00%
Storage				
NUCLEUS SIZE (V)	2352KB	2352KB	0KB	0.00%
TRACE TABLE (V)	400KB	400KB	0KB	0.00%
WKSET (V)	61	61	0	0.00%
PGBLPGS	15185	15176	-9	-0.06%
PGBLPGS/USER	8.3	8.3	0.0	-0.06%
FREEPGS	4779	4792	13	0.27%
FREE UTIL	0.96	0.96	0.00	-0.27%
SHRPGS	1051	1057	6	0.57%

Table 26 (Page 2 of 2). Use of virtual disks in storage as short term CMS T-disks / 9121-480 with 128MB of storage / 1KB vs. 4KB block size

Block Size **Release** **Run ID**	**1KB** **VM/ESA 2.1** **L26E1828**	**4KB** **VM/ESA 2.1** **L26E1826**	**Difference**	**% Difference**
Environment				
Real Storage	**96MB**	**96MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1820**	**1820**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Paging				
READS/SEC	1062	1108	46	4.33%
WRITES/SEC	821	866	45	5.48%
PAGE/CMD	29.627	31.174	1.547	5.22%
PAGE IO RATE (V)	348.800	365.900	17.100	4.90%
PAGE IO/CMD (V)	5.488	5.778	0.290	5.29%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	7.316	7.343	0.027	0.37%
Queues				
DISPATCH LIST	47.70	47.73	0.04	0.07%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	583	582	-1	-0.17%
VIO/CMD	9.173	9.191	0.018	0.20%
RIO RATE (V)	521	535	14	2.69%
RIO/CMD (V)	8.197	8.449	0.251	3.07%
NONPAGE RIO/CMD (V)	2.709	2.671	-0.038	-1.40%
MDC READS	351	349	-2	-0.57%
MDC WRITES	139	138	-1	-0.72%
MDC MODS	106	106	0	0.00%
MDC HIT RATIO	0.91	0.91	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	17.669	17.635	-0.034	-0.19%
DIAG/CMD	26.633	26.596	-0.037	-0.14%
DIAG 08/CMD	0.762	0.756	-0.006	-0.78%
DIAG 14/CMD	0.025	0.024	0.000	-1.75%
DIAG 58/CMD	1.251	1.251	0.000	-0.02%
DIAG 98/CMD	0.969	0.948	-0.021	-2.18%
DIAG A4/CMD	3.856	3.900	0.044	1.14%
DIAG A8/CMD	1.779	1.768	-0.011	-0.59%
DIAG 214/CMD	12.849	12.809	-0.040	-0.31%
SIE/CMD	58.703	59.047	0.344	0.59%
SIE INTCPT/CMD	36.983	37.200	0.217	0.59%
FREE TOTL/CMD	63.502	63.438	-0.065	-0.10%
VTAM Machines				
WKSET (V)	481	483	2	0.42%
TOT CPU/CMD (V)	3.8461	3.8516	0.0055	0.14%
CP CPU/CMD (V)	1.6521	1.6582	0.0061	0.37%
VIRT CPU/CMD (V)	2.1940	2.1934	-0.0006	-0.03%
DIAG 98/CMD (V)	0.969	0.948	-0.022	-2.23%

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

CMS Temporary Minidisks (Held Long Term)

In the FS7F CMS-intensive workload, each user′s B-disk is a R/W minidisk that starts out with no files but, as the workload progresses, files are copied into it, manipulated, and erased. The COBOL compiler also allocates its work files on the B-disk and its use of those workfiles accounts for much of the I/O activity to the B-disk. In total, the B-disk receives about 12% of the workload′s minidisk I/O activity.

In this section, measurements made with a virtual disk in storage substituted for the B-disk are compared to standard FS7F measurements. This is intended to represent the case where a virtual disk in storage is substituted for a temporary minidisk[6] that is acquired at logon and is never detached until the user logs off. The size of the virtual disk in storage was equivalent to 20 3390 cylinders. The measurements were obtained on a 9121-480 processor with 256MB of total storage.

This section also includes a comparison of 1KB-formatted to 4KB-formatted virtual disks in storage in this environment.

---

[6] Actually, the B-disk is an ordinary minidisk and not a T-disk. However, a minidisk and a T-disk provide essentially the same performance if they have matching characteristics (same size, device type, and so forth).

9121-480 / 256MB

Measurement Configuration: See page 99.

Measurement Discussion: Overall system performance decreased when virtual disks in storage were substituted for the B-disk. External response time (AVG LAST (T)) increased 23% (0.087 seconds) and the internal throughput rate (ITR (H)) decreased 1.9%.

Because they were held long-term, the impact of the virtual disks in storage on real storage requirements and paging was significant and overshadowed the benefits of decreased minidisk I/O. Page I/Os per command (PAGE IO/CMD (V)) increased by 1.10 while minidisk I/Os, as evidenced by NONPAGE RIO/CMD, decreased by 0.61 I/O per command. In addition, when temporary disk space is held long term, the workload benefits little from the speed and efficiency of the FORMAT command for virtual disks in storage.

One command, a COBOL compile, does a large number of I/Os to the B-disk. In spite of the strong trend towards increased external response time, this command showed an 18% response time improvement. This parallels the results discussed on page 104 for the 128MB T-disk substitution case. The virtual disk in storage benefits are experienced by the commands that make extensive use of it. Meanwhile, the side effects (resulting from the overall increase in real storage requirements) can increase the response times of the other commands.

Table 27 (Page 1 of 3). Use of virtual disks in storage as long term CMS T-disks / 9121-480 with 256MB of storage

Virtual Disk in Storage **Release** **Run ID**	**Not used** **VM/ESA 2.1** **L26E1868**	**B-disks** **VM/ESA 2.1** **L26E1866**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.123	0.141	0.018	14.63%
NONTRIV INT	0.453	0.519	0.066	14.57%
TOT INT	0.344	0.389	0.045	13.08%
TOT INT ADJ	0.303	0.353	0.049	16.23%
AVG FIRST (T)	0.247	0.310	0.064	25.71%
AVG LAST (T)	0.378	0.464	0.087	22.88%
Throughput				
AVG THINK (T)	26.29	26.26	-0.03	-0.13%
ETR	57.57	58.91	1.34	2.33%
ETR (T)	65.25	64.96	-0.29	-0.45%
ETR RATIO	0.882	0.907	0.025	2.79%
ITR (H)	73.02	71.60	-1.41	-1.94%
ITR	32.22	32.47	0.25	0.79%
EMUL ITR	47.55	48.97	1.42	2.98%
ITRR (H)	1.000	0.981	-0.019	-1.94%
ITRR	1.000	1.008	0.008	0.79%

Table 27 (Page 2 of 3). Use of virtual disks in storage as long term CMS T-disks / 9121-480 with 256MB of storage				
Virtual Disk in Storage **Release** **Run ID**	**Not used** **VM/ESA 2.1** **L26E1868**	**B-disks** **VM/ESA 2.1** **L26E1866**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Proc. Usage				
PBT/CMD (H)	27.391	27.932	0.541	1.98%
PBT/CMD	27.431	27.862	0.431	1.57%
CP/CMD (H)	9.416	10.007	0.592	6.28%
CP/CMD	8.888	9.390	0.502	5.64%
EMUL/CMD (H)	17.975	17.924	-0.050	-0.28%
EMUL/CMD	18.543	18.472	-0.071	-0.38%
Processor Util.				
TOTAL (H)	178.74	181.46	2.72	1.52%
TOTAL	179.00	181.00	2.00	1.12%
UTIL/PROC (H)	89.37	90.73	1.36	1.52%
UTIL/PROC	89.50	90.50	1.00	1.12%
TOTAL EMUL (H)	117.29	116.44	-0.85	-0.72%
TOTAL EMUL	121.00	120.00	-1.00	-0.83%
MASTER TOTAL (H)	89.04	90.40	1.36	1.53%
MASTER TOTAL	89.00	90.00	1.00	1.12%
MASTER EMUL (H)	52.67	51.70	-0.97	-1.84%
MASTER EMUL	54.00	54.00	0.00	0.00%
TVR(H)	1.52	1.56	0.03	2.26%
TVR	1.48	1.51	0.03	1.96%
Storage				
NUCLEUS SIZE (V)	2364KB	2364KB	0KB	0.00%
TRACE TABLE (V)	400KB	400KB	0KB	0.00%
WKSET (V)	82	70	-12	-14.63%
PGBLPGS	47770	19740	-28030	-58.68%
PGBLPGS/USER	25.7	10.6	-15.1	-58.68%
FREEPGS	4953	5211	258	5.21%
FREE UTIL	0.93	0.93	0.00	0.33%
SHRPGS	1134	1117	-17	-1.50%
Paging				
READS/SEC	625	809	184	29.44%
WRITES/SEC	422	583	161	38.15%
PAGE/CMD	16.045	21.427	5.382	33.55%
PAGE IO RATE (V)	170.200	240.600	70.400	41.36%
PAGE IO/CMD (V)	2.608	3.704	1.095	42.00%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.467	6.896	0.429	6.64%
Queues				
DISPATCH LIST	37.02	42.13	5.11	13.80%
ELIGIBLE LIST	0.00	0.02	0.02	na

Table 27 (Page 3 of 3). Use of virtual disks in storage as long term CMS T-disks / 9121-480 with 256MB of storage				
Virtual Disk in Storage **Release** **Run ID**	**Not used** **VM/ESA 2.1** **L26E1868**	**B-disks** **VM/ESA 2.1** **L26E1866**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
I/O				
VIO RATE	627	613	-14	-2.23%
VIO/CMD	9.608	9.436	-0.173	-1.80%
RIO RATE (V)	391	421	30	7.67%
RIO/CMD (V)	5.992	6.480	0.489	8.15%
NONPAGE RIO/CMD (V)	3.384	2.776	-0.608	-17.97%
MDC READS	361	342	-19	-5.26%
MDC WRITES	179	132	-47	-26.26%
MDC MODS	145	100	-45	-31.03%
MDC HIT RATIO	0.91	0.91	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	17.584	17.658	0.074	0.42%
DIAG/CMD	26.983	28.239	1.256	4.65%
DIAG 08/CMD	0.760	0.759	-0.001	-0.14%
DIAG 14/CMD	0.025	0.025	0.000	-0.49%
DIAG 58/CMD	1.251	1.250	0.000	-0.04%
DIAG 98/CMD	1.227	1.072	-0.156	-12.70%
DIAG A4/CMD	3.913	3.875	-0.038	-0.97%
DIAG A8/CMD	1.888	1.915	0.027	1.44%
DIAG 214/CMD	12.941	12.657	-0.284	-2.20%
SIE/CMD	54.586	57.447	2.861	5.24%
SIE INTCPT/CMD	37.664	39.064	1.400	3.72%
FREE TOTL/CMD	64.470	66.514	2.043	3.17%
VTAM Machines				
WKSET (V)	515	517	2	0.39%
TOT CPU/CMD (V)	3.8737	3.8141	-0.0596	-1.54%
CP CPU/CMD (V)	1.6431	1.6334	-0.0097	-0.59%
VIRT CPU/CMD (V)	2.2306	2.1807	-0.0499	-2.24%
DIAG 98/CMD (V)	1.227	1.072	-0.156	-12.69%
Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM				

9121-480 / 256MB / 1KB vs. 4KB Block Size

Measurement Configuration: See page 99.

Measurement Discussion: As was true for the 1KB versus 4KB comparison described on page 108, the results showed that the 4KB-formatted virtual disks in storage did not perform as well as the 1KB-formatted ones. External response time (AVG LAST (T)) increased by 10% (0.047 seconds) and the internal throughput rate (ITR (H)) decreased by 0.4%. NONPAGE RIO/CMD decreased by 0.05 while PAGE IO/CMD increased by 0.22.

Table 28 (Page 1 of 3). Use of virtual disks in storage as long term CMS T-disks / 9121-480 with 256MB of storage / 1KB vs. 4KB block size

Block Size **Release** **Run ID**	**1KB** **VM/ESA 2.1** **L26E1866**	**4KB** **VM/ESA 2.1** **L26E1865**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.141	0.147	0.006	4.26%
NONTRIV INT	0.519	0.562	0.043	8.29%
TOT INT	0.389	0.418	0.029	7.46%
TOT INT ADJ	0.353	0.382	0.029	8.36%
AVG FIRST (T)	0.310	0.337	0.027	8.53%
AVG LAST (T)	0.464	0.511	0.047	10.01%
Throughput				
AVG THINK (T)	26.26	26.29	0.03	0.11%
ETR	58.91	59.43	0.52	0.88%
ETR (T)	64.96	64.99	0.03	0.04%
ETR RATIO	0.907	0.914	0.008	0.84%
ITR (H)	71.60	71.29	-0.31	-0.44%
ITR	32.47	32.62	0.14	0.44%
EMUL ITR	48.97	49.32	0.35	0.71%
ITRR (H)	1.000	0.996	-0.004	-0.44%
ITRR	1.000	1.004	0.004	0.44%
Proc. Usage				
PBT/CMD (H)	27.932	28.054	0.123	0.44%
PBT/CMD	27.862	28.003	0.142	0.51%
CP/CMD (H)	10.007	10.099	0.091	0.91%
CP/CMD	9.390	9.386	-0.004	-0.04%
EMUL/CMD (H)	17.924	17.955	0.031	0.17%
EMUL/CMD	18.472	18.618	0.146	0.79%
Processor Util.				
TOTAL (H)	181.46	182.33	0.87	0.48%
TOTAL	181.00	182.00	1.00	0.55%
UTIL/PROC (H)	90.73	91.17	0.44	0.48%
UTIL/PROC	90.50	91.00	0.50	0.55%
TOTAL EMUL (H)	116.44	116.70	0.25	0.22%
TOTAL EMUL	120.00	121.00	1.00	0.83%
MASTER TOTAL (H)	90.40	90.82	0.41	0.46%
MASTER TOTAL	90.00	91.00	1.00	1.11%
MASTER EMUL (H)	51.70	51.78	0.08	0.15%
MASTER EMUL	54.00	54.00	0.00	0.00%
TVR(H)	1.56	1.56	0.00	0.26%
TVR	1.51	1.50	0.00	-0.28%

Table 28 (Page 2 of 3). Use of virtual disks in storage as long term CMS T-disks / 9121-480 with 256MB of storage / 1KB vs. 4KB block size				
Block Size	**1KB**	**4KB**		
Release	**VM/ESA 2.1**	**VM/ESA 2.1**		
Run ID	**L26E1866**	**L26E1865**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Storage				
NUCLEUS SIZE (V)	2364KB	2364KB	0KB	0.00%
TRACE TABLE (V)	400KB	400KB	0KB	0.00%
WKSET (V)	70	70	0	0.00%
PGBLPGS	19740	19771	31	0.16%
PGBLPGS/USER	10.6	10.6	0.0	0.16%
FREEPGS	5211	5205	-6	-0.12%
FREE UTIL	0.93	0.93	0.00	0.12%
SHRPGS	1117	1097	-20	-1.79%
Paging				
READS/SEC	809	848	39	4.82%
WRITES/SEC	583	619	36	6.17%
PAGE/CMD	21.427	22.572	1.145	5.34%
PAGE IO RATE (V)	240.600	254.900	14.300	5.94%
PAGE IO/CMD (V)	3.704	3.922	0.218	5.90%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.896	7.001	0.105	1.52%
Queues				
DISPATCH LIST	42.13	43.35	1.22	2.89%
ELIGIBLE LIST	0.02	0.00	-0.02	-100.00%
I/O				
VIO RATE	613	612	-1	-0.16%
VIO/CMD	9.436	9.417	-0.019	-0.21%
RIO RATE (V)	421	432	11	2.61%
RIO/CMD (V)	6.480	6.647	0.166	2.57%
NONPAGE RIO/CMD (V)	2.776	2.725	-0.051	-1.84%
MDC READS	342	342	0	0.00%
MDC WRITES	132	132	0	0.00%
MDC MODS	100	99	-1	-1.00%
MDC HIT RATIO	0.91	0.91	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	17.658	17.642	-0.016	-0.09%
DIAG/CMD	28.239	28.544	0.305	1.08%
DIAG 08/CMD	0.759	0.761	0.002	0.24%
DIAG 14/CMD	0.025	0.025	0.000	0.53%
DIAG 58/CMD	1.250	1.249	-0.001	-0.06%
DIAG 98/CMD	1.072	1.019	-0.052	-4.88%
DIAG A4/CMD	3.875	3.911	0.035	0.91%
DIAG A8/CMD	1.915	1.913	-0.002	-0.12%
DIAG 214/CMD	12.657	12.986	0.329	2.60%
SIE/CMD	57.447	57.730	0.283	0.49%
SIE INTCPT/CMD	39.064	38.679	-0.385	-0.98%
FREE TOTL/CMD	66.514	66.577	0.064	0.10%

Table 28 (Page 3 of 3). Use of virtual disks in storage as long term CMS T-disks / 9121-480 with 256MB of storage / 1KB vs. 4KB block size

Block Size **Release** **Run ID**	**1KB** **VM/ESA 2.1** **L26E1866**	**4KB** **VM/ESA 2.1** **L26E1865**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
VTAM Machines				
WKSET (V)	517	517	0	0.00%
TOT CPU/CMD (V)	3.8141	3.7868	-0.0273	-0.72%
CP CPU/CMD (V)	1.6334	1.6070	-0.0264	-1.62%
VIRT CPU/CMD (V)	2.1807	2.1798	-0.0009	-0.04%
DIAG 98/CMD (V)	1.072	1.020	-0.052	-4.87%

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

Additional Functional Enhancements

Improvements have been made that decrease the CP path length when simulating the loading of a wait state PSW. The effects of these improvements on a VSE V=R guest environment are discussed in "Load Wait State PSW Improvements" on page 119.

The size of the minidisk file cache has been made a parameter in DMSNGP. Using the FS7F0R workload, a study was done that varied the size of this cache. The results are discussed in "Adjusting the Minidisk File Cache Size" on page 122.

REXX SAA Level 2 architecture provides support for REXX stream I/O. This is a set of REXX input and output routines. CMS path length comparisons of these new routines were made with the corresponding CSL routines and EXECIO statements. The comparisons are discussed in "REXX SAA Level 2 Architecture" on page 126.

Load Wait State PSW Improvements

Workload: PACEX4

Hardware Configuration

Processor model:	9221-170
Processors used:	1
Storage	
Real:	128MB
Expanded:	128MB
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*VSAM*	*VSE Sys.*	*VM Sys.*
9336-20	6310	2				6	2	
9336-10	6310	1	1	1				1

Software Configuration

VSE version:	1.2.0
VSE supervisor mode:	370

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
VSE	1	VSE V=R	16MB/370	100		IOASSIST ON CCWTRANS OFF
VMMAP	1	Monitor	6MB/370	100		
SMART	1	RTM	6MB/370	100		
WRITER	1	CP monitor	6MB/XA	100		

Measurement Discussion: The measurements in this section illustrate the improvement made in CP′s simulation of loading a guest′s wait state PSW. This function has been handled entirely under SIE (Start Interpretive Execution) for guests that have dedicated processors on a complex with the PR/SM facility running in basic mode. Guests operating in environments other than this will benefit from this change. The PACE workload was chosen for its high I/O activity. Waiting for I/O to complete causes the VSE guest to frequently enter an enabled wait state. A V=R guest with dedicated DASD was the measured configuration that showed the most improvement from this item. The count of entries into guest enabled wait state can be found in the field GUESTWT, located in the RTM SYSTEM screen. This field was added to the SYSTEM screen as a modification to RTM Release 1.5.2; prior releases of RTM do not report this statistic. The amount of processor time saved is proportional to this number. In this evaluation the VSE guest entered an enabled wait state approximately 85 times per second.

In the following table the variable CP/CMD (H) showed a 4.6% decrease when comparing VM/ESA Release 2 to VM/ESA Release 2.1 due to the decreased overhead for this simulation. This decrease translated into a 2.3% increase in internal throughput rate (ITR (H)) for VM/ESA Release 2.1. Examination of the two VSE guest regression comparisons "9121-320 / V=V MODE=ESA" on page 57 and "9121-320 / V=V MODE=VMESA" on page 58, showed that these environments did not obtain as much benefit as did the V=R with dedicated DASD case. The V=V guests with non-dedicated DASD obtained less than a 1% increase in internal throughput rate.

Table 29 (Page 1 of 2). VM/ESA Release 2.1 load wait state PSW improvement

Release **Run ID**	**VM/ESA 2** **HB54RE31**	**VM/ESA 2.1** **HB64RE31**	**Difference**	**% Difference**
Environment				
IML Mode	**ESA**	**ESA**		
Real Storage	**128MB**	**128MB**		
Exp. Storage	**128MB**	**128MB**		
VM Mode	**370**	**370**		
VM Size	**16MB**	**16MB**		
Guest Setting	**V=R**	**V=R**		
VSE Supervisor	**370**	**370**		
Processors	**1**	**1**		
Throughput (Min)				
Elapsed Time (H)	850.7	854.5	3.9	0.45%
ETR (H)	1.97	1.97	-0.01	-0.45%
ITR (H)	5.41	5.54	0.12	2.29%
ITR	5.49	5.46	-0.02	-0.45%
ITRR (H)	1.000	1.023	0.023	2.29%
ITRR	1.000	0.995	-0.005	-0.45%
Proc. Usage (Sec)				
PBT/CMD (H)	11.084	10.836	-0.248	-2.24%
PBT/CMD	10.937	10.987	0.050	0.45%
CP/CMD (H)	5.167	4.928	-0.239	-4.63%
CP/CMD	3.950	3.662	-0.287	-7.27%
EMUL/CMD (H)	5.917	5.909	-0.009	-0.15%
EMUL/CMD	6.988	7.325	0.337	4.82%
Processor Util.				
TOTAL (H)	36.48	35.51	-0.98	-2.68%
TOTAL	36.00	36.00	0.00	0.00%
TOTAL EMUL (H)	19.48	19.36	-0.12	-0.60%
TOTAL EMUL	23.00	24.00	1.00	4.35%
TVR(H)	1.87	1.83	-0.04	-2.09%
TVR	1.57	1.50	-0.07	-4.17%
Storage				
NUCLEUS SIZE (V)	2460KB	2520KB	60KB	2.44%
TRACE TABLE (V)	1492KB	1492KB	0KB	0.00%
PGBLPGS	22464	22455	-9	-0.04%
FREEPGS	74	73	-1	-1.35%
FREE UTIL	0.48	0.49	0.01	1.37%
SHRPGS	1007	1007	0	0.00%
Paging				
PAGE/CMD	0.000	0.000	0.000	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	0.000	0.000	0.000	na
I/O				
VIO RATE	0.000	0.000	0.000	na
VIO/CMD	0.000	0.000	0.000	na
RIO RATE (V)	1.000	1.000	0.000	0.00%
RIO/CMD (V)	30.381	30.519	0.138	0.45%
DASD IO TOTAL (V)	94431	93946	-485	-0.51%
DASD IO RATE (V)	112.42	111.84	-0.58	-0.51%
DASD IO/CMD (V)	3415.40	3413.30	-2.10	-0.06%

Table 29 (Page 2 of 2). VM/ESA Release 2.1 load wait state PSW improvement

Release **Run ID**	**VM/ESA 2** **HB54RE31**	**VM/ESA 2.1** **HB64RE31**	**Difference**	**% Difference**
Environment				
IML Mode	**ESA**	**ESA**		
Real Storage	**128MB**	**128MB**		
Exp. Storage	**128MB**	**128MB**		
VM Mode	**370**	**370**		
VM Size	**16MB**	**16MB**		
Guest Setting	**V=R**	**V=R**		
VSE Supervisor	**370**	**370**		
Processors	**1**	**1**		
PRIVOPs				
PRIVOP/CMD	12.764	12.222	-0.542	-4.25%
DIAG/CMD	1878.084	1916.105	38.021	2.02%
SIE/CMD	5012.913	5127.251	114.338	2.28%
SIE INTCPT/CMD	4561.750	4665.798	104.048	2.28%
FREE TOTL/CMD	9661.250	9918.788	257.539	2.67%
GUESTWT/CMD	2641.964	2651.464	-9.5	-0.36%

Note: V=VMPRF, H=Hardware Monitor, Unmarked=RTM

Adjusting the Minidisk File Cache Size

Workload: FS7F0R

Hardware Configuration

Processor model:	9121-480
Processors used:	2
Storage	
Real:	128MB
Expanded:	None
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-3	2						2 R
3390-2	3990-2	4	16	5	6			
3390-2	3990-3	4		2	2	16 R		

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
SMART	1	RTM	16MB/370	3%	400	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	480	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	1820	User	3MB/XC	100		

Measurement Discussion:

This section documents the results of changing the CMS minidisk file cache size when running VM/ESA Release 2.1 on a 9121-480.

The minidisk file cache is used to reduce the frequency of minidisk file I/Os. When a file is read sequentially, CMS reads as many blocks at a time as will fit into the minidisk file cache. When a file is written sequentially, completed blocks are accumulated until they fill the file cache and are then written together.

In prior releases, the size of this cache was fixed at 8KB. This remains the default, but the size can now be set to a value from 1KB to 96KB when the CMS nucleus is built.

As the file cache size increases, minidisk I/Os decrease, but system real storage requirements and paging may increase. The 8KB default is suitable for most storage constrained environments, while a larger size is likely to result in an overall performance improvement for systems with more real storage.

The following table documents the results of using a 4KB, 8KB, 16KB, and 96KB cache for the FS7F workload, in a moderately storage constrained environment. This environment had no expanded storage. Environments with expanded storage used for minidisk cache (MDC) or paging (or both) could see different results.

As the file cache size increases, the overall external response times (AVG LAST (T)) decreased, except in the largest case measured (96KB). The internal throughput rates (ITR (H)) also improved due to the reduced minidisk I/O rate (DIAG A4/CMD). This environment showed no significant change in paging rate (PAGE/CMD), except in the 96KB measurement.

The file cache size represents a trade-off between minidisk I/Os and paging. In this comparison however, the increased file cache size did not drive up the paging rate until the 96KB case. Apparently, the increased number of referenced buffer pages was balanced by the fact that they were held for a shorter time because the I/Os were done more efficiently. The virtual pages used are obtained at open time as first reference pages and released after the file is closed. There is little chance of these pages being selected for paging out to expanded storage or DASD.

The data shown in the table for the 96KB measurement does not explain why the external response time increased. The increase in paging rate should have been offset by the decrease in minidisk I/O rate. Looking closer at the DASD response times (data not shown), indicated that the time to execute the longer channel programs for minidisk I/O increased. Therefore, even though the DASD rates indicated there should be an improvement, the actual minidisk DASD time remained about the same. This, plus the fact that paging increased, caused the net external response time to increase.

Table 30 (Page 1 of 2). Minidisk file cache size on VM/ESA Release 2.1 on the 9121-480

Cache Size **Release** **Run ID**	**4KB** **VM/ESA 2.1** **L26E1823**	**8KB** **VM/ESA 2.1** **L26E1821**	**16KB** **VM/ESA 2.1** **L26E1822**	**96KB** **VM/ESA 2.1** **L26E182C**
Environment				
Real Storage	**128MB**	**128MB**	**128MB**	**128MB**
Exp. Storage	**None**	**None**	**None**	**None**
Users	**1820**	**1820**	**1820**	**1820**
VTAMs	**1**	**1**	**1**	**1**
VSCSs	**0**	**0**	**0**	**0**
Processors	**2**	**2**	**2**	**2**
Response Time				
TRIV INT	0.133	0.133	0.131	0.143
NONTRIV INT	0.533	0.500	0.477	0.528
TOT INT	0.402	0.379	0.363	0.401
TOT INT ADJ	0.355	0.334	0.321	0.357
AVG FIRST (T)	0.254	0.255	0.259	0.283
AVG LAST (T)	0.428	0.410	0.402	0.447
Throughput				
AVG THINK (T)	26.31	26.36	26.31	26.32
ETR	56.15	56.39	56.46	56.76
ETR (T)	63.66	63.90	63.84	63.74
ETR RATIO	0.882	0.882	0.884	0.890
ITR (H)	70.35	71.70	72.36	72.38
ITR	31.05	31.67	32.01	32.25
EMUL ITR	46.14	46.52	46.78	47.06
ITRR (H)	1.000	1.019	1.029	1.029
ITRR	1.000	1.020	1.031	1.039
Proc. Usage				
PBT/CMD (H)	28.430	27.893	27.638	27.631
PBT/CMD	28.430	27.857	27.570	27.612
CP/CMD (H)	10.054	9.651	9.476	9.488
CP/CMD	9.267	8.920	8.616	8.629
EMUL/CMD (H)	18.375	18.242	18.163	18.143
EMUL/CMD	19.163	18.936	18.954	18.983
Processor Util.				
TOTAL (H)	181.00	178.24	176.44	176.12
TOTAL	181.00	178.00	176.00	176.00
UTIL/PROC (H)	90.50	89.12	88.22	88.06
UTIL/PROC	90.50	89.00	88.00	88.00
TOTAL EMUL (H)	116.99	116.57	115.95	115.64
TOTAL EMUL	122.00	121.00	121.00	121.00
MASTER TOTAL (H)	90.67	89.18	88.29	88.00
MASTER TOTAL	91.00	89.00	88.00	88.00
MASTER EMUL (H)	53.43	52.81	52.45	51.96
MASTER EMUL	56.00	55.00	55.00	54.00
TVR(H)	1.55	1.53	1.52	1.52
TVR	1.48	1.47	1.45	1.45
Storage				
NUCLEUS SIZE (V)	2352KB	2352KB	2352KB	2352KB
TRACE TABLE (V)	400KB	400KB	400KB	400KB
WKSET (V)	78	78	79	77
PGBLPGS	23335	23335	23342	23290
PGBLPGS/USER	12.8	12.8	12.8	12.8
FREEPGS	5052	5050	5046	5069
FREE UTIL	0.96	0.96	0.96	0.96
SHRPGS	1101	1081	1046	1076

Table 30 (Page 2 of 2). Minidisk file cache size on VM/ESA Release 2.1 on the 9121-480

Cache Size Release Run ID	4KB VM/ESA 2.1 L26E1823	8KB VM/ESA 2.1 L26E1821	16KB VM/ESA 2.1 L26E1822	96KB VM/ESA 2.1 L26E182C
Environment				
Real Storage	**128MB**	**128MB**	**128MB**	**128MB**
Exp. Storage	**None**	**None**	**None**	**None**
Users	**1820**	**1820**	**1820**	**1820**
VTAMs	**1**	**1**	**1**	**1**
VSCSs	**0**	**0**	**0**	**0**
Processors	**2**	**2**	**2**	**2**
Paging				
READS/SEC	659	674	672	693
WRITES/SEC	454	458	455	473
PAGE/CMD	17.482	17.716	17.654	18.293
PAGE IO RATE (V)	194.800	194.800	192.100	203.100
PAGE IO/CMD (V)	3.060	3.049	3.009	3.186
XSTOR IN/SEC	0	0	0	0
XSTOR OUT/SEC	0	0	0	0
XSTOR/CMD	0.000	0.000	0.000	0.000
FAST CLR/CMD	6.503	6.635	6.845	7.703
Queues				
DISPATCH LIST	45.25	42.05	39.77	45.67
ELIGIBLE LIST	0.00	0.00	0.00	0.00
I/O				
VIO RATE	695	611	563	515
VIO/CMD	10.917	9.562	8.819	8.080
RIO RATE (V)	665	584	534	499
RIO/CMD (V)	10.445	9.139	8.365	7.829
MDC READS	0	0	0	0
MDC WRITES	0	0	0	0
MDC MODS	0	0	0	0
MDC HIT RATIO	0.00	0.00	0.00	0.00
PRIVOPs				
PRIVOP/CMD	17.571	17.541	17.514	17.528
DIAG/CMD	28.354	26.833	26.120	25.380
DIAG 08/CMD	0.762	0.760	0.761	0.760
DIAG 14/CMD	0.025	0.025	0.025	0.025
DIAG 58/CMD	1.252	1.253	1.253	1.252
DIAG 98/CMD	1.221	1.176	1.203	1.158
DIAG A4/CMD	5.210	3.917	3.151	2.456
DIAG A8/CMD	1.906	1.888	1.891	1.883
DIAG 214/CMD	12.929	12.816	12.793	12.805
SIE/CMD	60.049	57.544	56.487	56.197
SIE INTCPT/CMD	39.032	37.979	37.281	36.528
FREE TOTL/CMD	76.432	68.734	66.074	62.503
VTAM Machines				
WKSET (V)	484	485	486	498
TOT CPU/CMD (V)	3.9879	3.9472	3.9510	3.9047
CP CPU/CMD (V)	1.6842	1.6693	1.6883	1.6560
VIRT CPU/CMD (V)	2.3037	2.2779	2.2627	2.2487
DIAG 98/CMD (V)	1.222	1.175	1.203	1.159

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

REXX SAA Level 2 Architecture

This item provides support for REXX stream I/O—a set of REXX input and output routines. Previously, REXX applications had to rely on VM/ESA services such as CMS EXECIO to do I/O. Now VM/ESA REXX applications can use REXX stream I/O and be consistent with REXX applications on other SAA* platforms.

A CMS path length comparison was made between three I/O methods. Code fragments for reading records follows (the write tests were similar):

1. CSL (was opened with intent to read):

```
call csl 'DMSOPEN cslrc cslrsn fnftfm fnftfmL intent intentL token'
call csl 'DMSREAD cslrc cslrsn token NumR RL Buf BufL BytR'
call csl 'DMSCLOSE cslrc cslrsn token commit commitL'
```

2. EXECIO (the first EXECIO has an implicit open and reads the first record; the FINIS closes the file):

```
'EXECIO 1 DISKR' fnftfm '(VAR X'
'EXECIO 1 DISKR' fnftfm '(VAR X'
'FINIS' fnftfm
```

3. Stream I/O:

```
parse value stream(fnftfm,'c','open read') with ok handle
y = linein(handle)
y= stream(handle,'C','CLOSE')
```

The following table summarizes the path lengths measured:

Table 31. Path length comparison between CSL, EXECIO, and Stream I/O.

Function	CSL	EXECIO	Stream I/O
OPEN (read)	15134	* 6921	28341
READ	13520	5460	4131
CLOSE	12399	2885	5096
OPEN (write)	14934	* 8306	28228
WRITE	12234	5751	4559
CLOSE	14324	4810	7028
TOTAL	82545	34133	77383

Note: * This OPEN includes the reading/writing of the first record.

Stream I/O was slightly faster than CSL in the above scenario (77383 versus 82545). If more records were read or written, the stream I/O advantage would increase because each read was 69% faster with stream I/O and each write was 62% faster.

Stream I/O was slower than EXECIO (77383 versus 34133). If enough records were read or written, stream I/O could be faster because, for each read, stream I/O was 24% faster than EXECIO and each write was 20% faster. However, if there are multiple records, EXECIO does have the capability to read or write multiple records using a stem variable; this would give the advantage back to EXECIO.

Additional Evaluations

This chapter explores some additional performance-related topics.

- "Processor Capacity" on page 128 explains measurement results from running a CMS-intensive workload with VM/ESA Release 2.1 on a 9121-742 varying the numbers of processors.
- "VSE/ESA Guest Performance" on page 135 summarizes the results from running VSE/ESA as a guest of VM/ESA Release 2.1.
- "3745 Comparison to CTCA" on page 137 compares using a 3088 CTCA to a 3745 device for VTAM network connectivity.
- "CP Monitor Overhead" on page 142 discusses the cost of collecting CP monitor data.
- "Minidisk Cache Tuning: Restricting the Arbiter" on page 146 is an example of using the CP RETAIN XSTORE command to improve overall system performance.
- "High Level Assembler Evaluation" on page 151 explains measurement results from running a CMS-intensive workload with VM/ESA Release 2.1 on a 9121-480 varying the assembler from HASM to HLASM.
- "VM/ESA REXX Performance Guidelines" on page 155 is a collection of performance guidelines that can be used when coding REXX programs in CMS.
- "Measurement Variability" on page 180 presents data showing how much variability in measurement results can be expected on a 9121-480 processor using a CMS-intensive workload.

Processor Capacity

The processor capacity measurements were made to determine the performance of VM/ESA Release 2.1 when running on different numbers of processors on a 9121-742. The performance of VM/ESA Release 2.1 scaled as expected in measurements made on the 9121-742 configured with 1, 2, 3, and 4 processors online. The measurements were done to get insight into how VM/ESA Release 2.1 would perform on different size processors within the 9121 511-based family. The results are very similar to earlier measurements made on the 9121 511-based processors with VM/ESA Release 1.1 using the LSPR workloads.[7]

The following graph represents the internal throughput rate (ITR) as a function of the number of processors online for the 9121-742. The dotted line, shown for reference, is a linear projection assuming no loss of ITR going from 1 to 4 processors online. The graph shows that VM/ESA Release 2.1 performs well as the number of processors is increased.

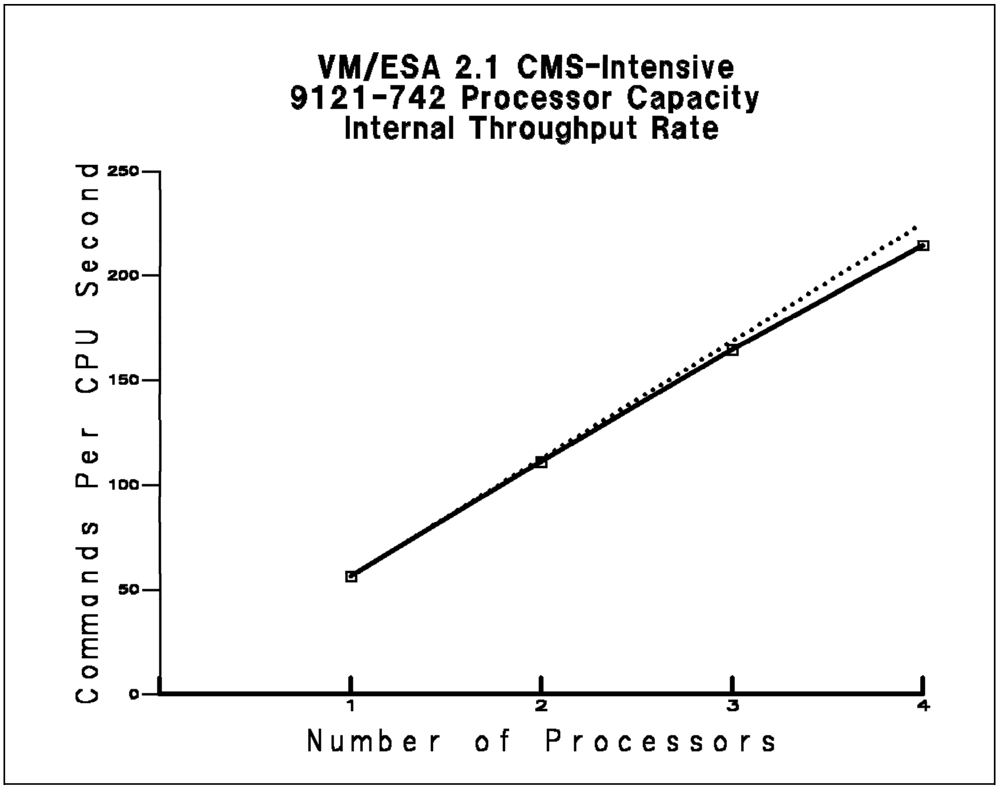

Figure 14. Internal throughput rate for the 9121-742 processor.

[7] The LSPR workloads for VM, PD4 and HT5, were developed for the purpose of comparing processors. Their key metric is ITR, with no constraints existing on storage or DASD. LSPR is an abbreviation for Large System Performance Reference.

Workload: FS7F0R

Hardware Configuration

Processor Model	*Processors Used*	*Real Storage*	*Expanded Storage*	*Retained for MDC*
9121-742	1	480MB	32MB	All
9121-742	2	992MB	32MB	All
9121-742	3	1024MB	512MB	64MB - 512MB
9121-742	4	1024MB	1024MB	64MB - 1024MB

Tape: 3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes -* *TDSK*	*User*	*Server*	*System*
3390-2	3990-2	2	6	4				
3390-2	3990-2	4	16	5	6			
3390-2	3990-3	4		2	7	8 R per processor		2 R

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Note: For each of the measurements, all of the DASD volumes were online. Because of the reduced number of users run when fewer processors were online, not all of the DASD volumes for SPOOL and minidisks were required or used.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
For the 9121-742 (1-Way):						
SMART	1	RTM	16MB/370	3%	500	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	575	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	1440	User	3MB/XC	100		
For the 9121-742 (2-Way):						
SMART	1	RTM	16MB/370	3%	500	QUICKDSP ON
VSCS1	1	VSCS	64MB/XA	10000	1200	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	590	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	2850	User	3MB/XC	100		
For the 9121-742 (3-Way):						
SMART	1	RTM	16MB/370	3%	500	QUICKDSP ON
VSCSn	2	VSCS	64MB/XA	10000	1200	QUICKDSP ON
VTAMXA	1	VTAM	64MB/XA	10000	550	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	4220	User	3MB/XC	100		
For the 9121-742 (4-Way):						
SMART	1	RTM	16MB/370	3%	500	QUICKDSP ON
VSCSn	3	VSCS	64MB/XA	10000	1200	QUICKDSP ON
VTAMXA	1	VTAM	64MB/XA	10000	550	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	5500	User	3MB/XC	100		

Measurement Discussion: The processor capacity measurements were made using a 9121-742 processor. A processor utilization of 90% was chosen as the target for the measurements. The VARY OFFLINE PROCESSORS command was used to disable excess processors.

For each measurement, the 9121-742 (4-way) was configured for the selected real storage and expanded storage sizes. To configure 32MB of expanded storage for the 1-way and 2-way measurements, 512MB (plus 32MB from real for the 1-way) was configured as expanded storage on side 0, and the excess was attached to the operator. The goal in selecting storage sizes was for the sum of real and expanded storage to scale with the number of processors. For the 9121 processors there is no dedicated expanded storage; total storage is configured into increments of central and expanded storage at IML, with 1024MB as the maximum amount of central storage. For the 1-way and 2-way measurements, all of the required storage fit below the 1024MB line, so it was configured as mostly central storage with 32MB defined for exclusive use for minidisk cache. The 3-way and 4-way measurements reserved a minimum 64MB of expanded storage for minidisk caching. Previous experiments had determined that this amount is needed on these systems for a good minidisk cache hit ratio. Refer to “Minidisk Cache Tuning: Restricting the Arbiter” on page 146 for more information.

While each measurement had a single VTAM, the number and type of VSCSs active as VTAM Service Machines (VSMs) varied, as did the number of TPNS machines. The 1-way measurement had an internal VSCS (within the VTAM virtual machine) to handle the user traffic driven by one TPNS machine. The 2-way measurement used both an internal VSCS and an external VSCS machine, with user traffic driven by two TPNS machines. The 3-way and 4-way measurements did not use the internal VSCS. These measurements had external VSCS and TPNS machines equal to the number of processors minus 1. The numbers and types of VSCS machines were chosen so that there would be one processor for each of the VTAM or VSCS machines needing to be dispatched.

Table 33 on page 133 summarizes the four processor capacity measurements. The results show that VM/ESA Release 2.1 scaled as expected on the 9121-742 as the number of processors was increased from 1 to 4. The ITRs ranged from 56.3 on the 1-way to 214.4 for the 4-way with ITRRs of 1, 1.97, 2.92, and 3.81. The following table shows that these results were similar to studies of VM/ESA Release 1.1 on the 9121 511-based family using the LSPR workloads.

Table 32. ITR and ITR ratios for FS7F workload on VM/ESA Release 2.1 compared to PD4 and HT5 LSPR workloads on VM/ESA Release 1.1 on the 9121-742.

Variable	Workload	1-way	2-way	3-way	4-way
ITR(H)	FS7F	56.29	111.05	164.54	214.45
	PD4	33.41	65.50	96.94	129.59
	HT5	80.59	153.64	228.58	299.32
ITRR(H)	FS7F	1.000	1.973	2.923	3.810
	PD4	1.000	1.961	2.901	3.879
	HT5	1.000	1.907	2.836	3.714
ITRR(H) per processor	FS7F	1.000	.986	.974	.952
	PD4	1.000	.980	.967	.970
	HT5	1.000	.954	.945	.928

The decrease in ITRR/processor is mostly caused by normal costs of inter-processor communications. There are additional processing requirements generated by using external VSCS machines. They are largely offset, however, by the decrease in DIAGNOSE code X′98′ instructions issued per command as the system size increased. This reflects an increase in VTAM′s ability to chain I/O buffers and accounts for the decrease in VIRT CPU/CMD (V) for the VTAM machines.

The trade-off between the effects of 1) the decrease in DIAGNOSE code X′98′ instructions (which generates less IUCV activity) and 2) the increased use of external VSCS machines (which generates more IUCV activity) also accounts for the increases in PRIVOP/CMD. From the 1-way to the 2-way to the 3-way measurement, the second factor exists and wins out over the first. From the 3-way to the 4-way measurement, there is no change from internal to external VSCS machines, and factor 2 drops out. Factor 1 explains the drop in PRIVOP/CMD from the 3-way to the 4-way measurement.

There was a 19% increase in response time (AVG LAST (T)) from the 3-way to the 4-way measurement. This is not expected to have any significant impact on ITR for this environment.

The decrease in DIAG/CMD is mostly accounted for by the decreases in DIAGNOSE code X′04′ and DIAGNOSE code X′98′ instructions issued per command. The DIAGNOSE code X′04′ instruction is issued by RTM to collect data for devices, users, and other system data. For any given interval of time, the number of DIAGNOSE code X′04′ instructions issued is constant because the same IOCP was used for each of the measurements and because the RTM sampling interval is constant. RTM issues the DIAGNOSE code X′04′ instruction twice for each online device and once for each offline device at each computation interval. The DIAGNOSE instruction is also issued once per VMDBK and once per ASCBK, up to the user table limit maximum (set at 2000 for these measurements). For any given interval of time, more commands are executed as we move from the 1-way to the 4-way measurements because of the increased number of users. The constant rate of DIAGNOSE code X′04′ instructions gets prorated among more and more commands from the 1-way to the 4-way measurements, resulting in the per-command decrease seen in the following table.

Table 33 (Page 1 of 2). VM/ESA Release 2.1 on the 9121-742 using different numbers of processors.

Processors **Release** **Run ID**	**1** **VM/ESA 2.1** **S16E1440**	**2** **VM/ESA 2.1** **S26E2850**	**3** **VM/ESA 2.1** **S36E4220**	**4** **VM/ESA 2.1** **S46E5500**
Environment				
Real Storage	**480MB**	**992MB**	**1024MB**	**1024MB**
Exp. Storage	**32MB**	**32MB**	**512MB**	**1024MB**
Users	**1440**	**2850**	**4220**	**5500**
VTAMs	**1**	**1**	**1**	**1**
VSCSs	**0**	**1**	**2**	**3**
Response Time				
TRIV INT	0.083	0.101	0.102	0.104
NONTRIV INT	0.393	0.365	0.363	0.372
TOT INT	0.301	0.279	0.272	0.267
TOT INT ADJ	0.258	0.243	0.245	0.258
AVG FIRST (T)	0.175	0.195	0.210	0.268
AVG LAST (T)	0.296	0.292	0.305	0.365
Throughput				
AVG THINK (T)	26.32	26.30	26.28	26.20
ETR	43.45	87.36	133.56	186.42
ETR (T)	50.63	100.21	148.31	192.70
ETR RATIO	0.858	0.872	0.901	0.967
ITR (H)	56.29	111.05	164.54	214.45
ITR	48.34	48.46	49.45	51.91
EMUL ITR	69.18	69.85	73.31	78.20
ITRR (H)	1.000	1.973	2.923	3.810
ITRR	1.000	1.002	1.023	1.074
Proc. Usage				
PBT/CMD (H)	17.765	18.011	18.233	18.652
PBT/CMD	17.776	17.963	18.205	18.630
CP/CMD (H)	5.703	5.886	6.312	6.658
CP/CMD	5.333	5.489	5.933	6.227
EMUL/CMD (H)	12.062	12.125	11.920	11.994
EMUL/CMD	12.443	12.474	12.272	12.403
Processor Util.				
TOTAL (H)	89.94	180.48	270.41	359.44
TOTAL	90.00	180.00	270.00	359.00
UTIL/PROC (H)	89.94	90.24	90.14	89.86
UTIL/PROC	90.00	90.00	90.00	89.75
TOTAL EMUL (H)	61.07	121.50	176.79	231.13
TOTAL EMUL	63.00	125.00	182.00	239.00
MASTER TOTAL (H)	89.94	90.21	91.03	92.02
MASTER TOTAL	90.00	90.00	91.00	92.00
MASTER EMUL (H)	61.07	54.61	47.65	41.75
MASTER EMUL	63.00	56.00	49.00	43.00
TVR(H)	1.47	1.49	1.53	1.56
TVR	1.43	1.44	1.48	1.50
Storage				
NUCLEUS SIZE (V)	2368KB	2368KB	2368KB	2368KB
TRACE TABLE (V)	200KB	400KB	600KB	800KB
WKSET (V)	88	88	73	69
PGBLPGS	114 k	239 k	241 k	236 k
PGBLPGS/USER	79.2	83.9	57.1	42.9
FREEPGS	3961	7754	11459	14829
FREE UTIL	0.93	0.92	0.92	0.91
SHRPGS	1040	1231	1428	1553

Table 33 (Page 2 of 2). VM/ESA Release 2.1 on the 9121-742 using different numbers of processors.

Processors **Release** **Run ID**	**1** **VM/ESA 2.1** **S16E1440**	**2** **VM/ESA 2.1** **S26E2850**	**3** **VM/ESA 2.1** **S36E4220**	**4** **VM/ESA 2.1** **S46E5500**
Environment				
Real Storage	**480MB**	**992MB**	**1024MB**	**1024MB**
Exp. Storage	**32MB**	**32MB**	**512MB**	**1024MB**
Users	**1440**	**2850**	**4220**	**5500**
VTAMs	**1**	**1**	**1**	**1**
VSCSs	**0**	**1**	**2**	**3**
Paging				
READS/SEC	411	748	495	556
WRITES/SEC	257	509	297	345
PAGE/CMD	13.194	12.544	5.340	4.676
PAGE IO RATE (V)	96.000	195.300	107.800	126.000
PAGE IO/CMD (V)	1.896	1.949	0.727	0.654
XSTOR IN/SEC	0	0	618	892
XSTOR OUT/SEC	0	0	1014	1353
XSTOR/CMD	0.000	0.000	11.004	11.650
FAST CLR/CMD	6.301	6.447	6.534	6.528
Queues				
DISPATCH LIST	27.00	52.09	75.79	107.24
ELIGIBLE LIST	0.00	0.00	0.00	0.00
I/O				
VIO RATE	518	940	1318	1679
VIO/CMD	10.231	9.381	8.887	8.713
RIO RATE	303	505	490	586
RIO/CMD	5.985	5.040	3.304	3.041
MDC READS	280	554	821	1067
MDC WRITES	138	274	406	528
MDC MODS	112	223	330	429
MDC HIT RATIO	0.91	0.91	0.91	0.91
PRIVOPs				
PRIVOP/CMD	17.431	19.914	21.149	20.437
DIAG/CMD	29.098	26.647	25.370	24.833
DIAG 04/CMD	3.900	2.295	1.498	1.133
DIAG 08/CMD	0.762	0.760	0.761	0.761
DIAG 14/CMD	0.025	0.025	0.025	0.025
DIAG 58/CMD	1.250	1.250	1.250	1.251
DIAG 98/CMD	1.844	1.014	0.505	0.328
DIAG A4/CMD	3.913	3.908	3.896	3.892
DIAG A8/CMD	1.906	1.886	1.894	1.896
DIAG 214/CMD	12.948	12.966	12.998	13.002
SIE/CMD	56.330	53.718	53.927	51.894
SIE INTCPT/CMD	41.684	39.214	37.749	35.288
FREE TOTL/CMD	67.549	64.815	64.621	62.273
VTAM Machines				
WKSET (V)	575	1605	2950	4144
TOT CPU/CMD (V)	2.7432	2.7110	2.5622	2.6091
CP CPU/CMD (V)	1.0863	1.1532	1.2062	1.2483
VIRT CPU/CMD (V)	1.6569	1.5579	1.3560	1.3608
DIAG 98/CMD (V)	1.844	1.014	0.504	0.328

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

VSE/ESA Guest Performance

The guest performance for VM/ESA Release 2.1 is summarized in the charts for the PACEX8 workload for the 9121-320 and 9221-170 processors. The V=R guest performance is normalized to a base value of 1.00 in each illustration.

Note:

- The graphs were built using data obtained from the sections listed under “VSE/ESA Guest” on page 54.
- The V=R guest with dedicated devices is the best-case performer when VSE is run as a guest of VM/ESA.
- The V=V guest DASD type (dedicated or minidisk), does not affect performance.
- For workloads that are not as I/O intensive as the PACE workload, the internal throughput ratios for the various guest environments would be closer to 1.00.

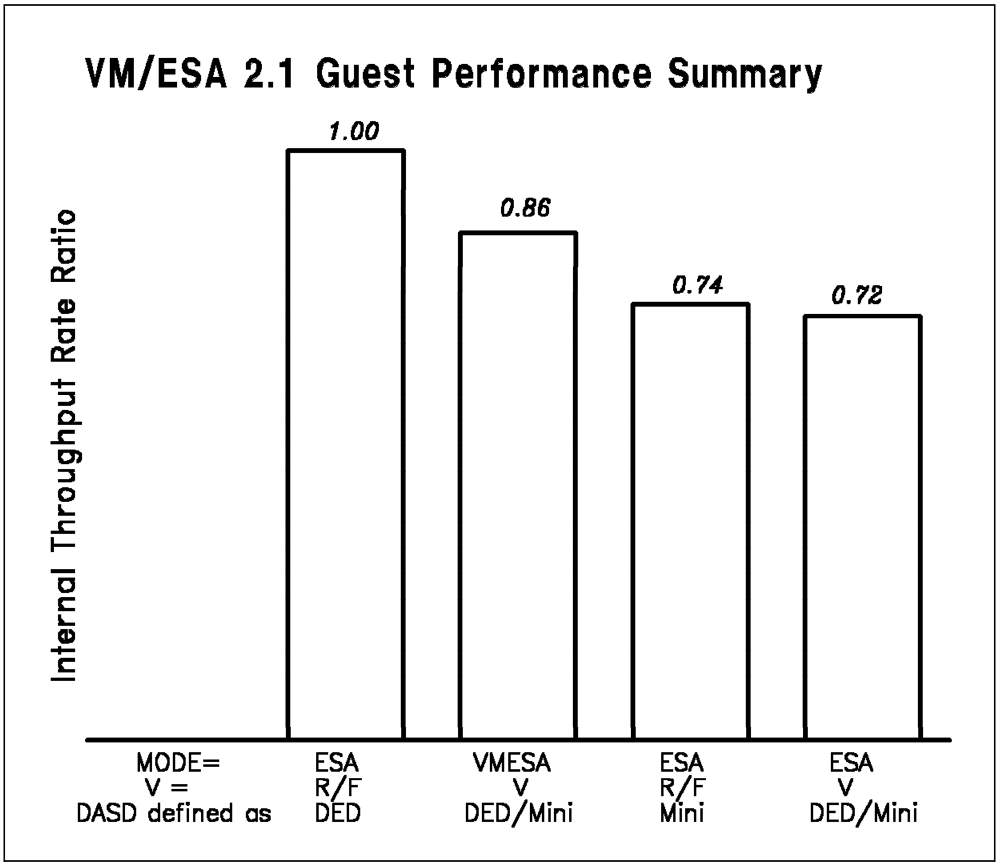

Figure 15. VM/ESA Release 2.1 guest performance summary. PACEX8 workload on the 9121-320 processor.

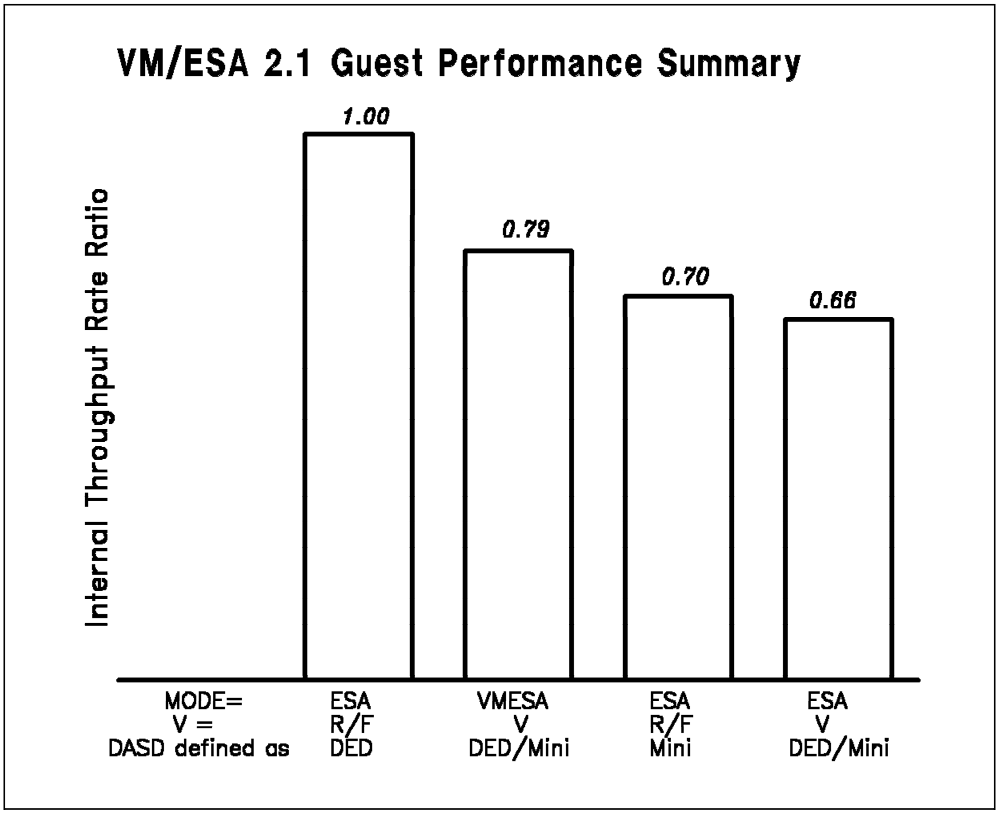

Figure 16. VM/ESA Release 2.1 guest performance summary. PACEX8 workload on the 9221-170 processor.

3745 Comparison to CTCA

Most of the CMS-intensive measurements provided in this report were done with a 3088 channel-to-channel adapter (CTCA) providing the network connectivity, while others (the SFS measurements on the 9121-480) used a 3745 control unit. This section presents and discusses the results of a comparable pair of measurements on VM/ESA Release 2.1 where one measurement was with CTCA connectivity and the other used a 3745.

Workload: FS7FMAXR

Hardware Configuration

Processor model:	9121-480
Processors used:	2
Storage	
Real:	224MB
Expanded:	32MB (all for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-2	4	16	4	6			
3390-2	3990-3	2						2 R
3390-2	3990-3	4		2	2		16 R	

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08 or	1	NA	4.5MB
3745-410	1	42	56Kb

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
CRRSERV1	1	SFS	16MB/XC	100		
ROSERV1	1	SFS	32MB/XC	100		QUICKDSP ON
RWSERVn	2	SFS	64MB/XC	1500	1200 (CTCA) 1300 (3745)	QUICKDSP ON
SMART	1	RTM	16MB/370	3%	300	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	450 (CTCA) 600 (3745)	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	1620	User	3MB/XC	100		

Measurement Discussion: The number of users was adjusted so that the two measurements were done at approximately the same processor utilization (88%).

The number of pages reserved for the VTAM machine was increased from 450 in the CTCA case to 600 in the 3745 case in order to accommodate the larger VTAM working set size that occurs in the 3745 case. The number of pages that was reserved for the SFS server virtual machines was slightly different (1200 and 1300) for the two measurements but the measurement data indicate that this had no significant effect on the results.

As expected, the two cases showed somewhat different performance characteristics. Relative to the CTCA measurement, the 3745 measurement had a 30% higher (0.136 seconds) external response time (AVG LAST (T)), while the internal throughput rate (ITR (H)) improved by 1.8%.

The response time increase is probably due to the reduced bandwidth of the 3745 (56Kb) relative to the CTCA (4.5MB). However, the details of how VTAM and the 3745 were configured may be a factor as well.

Some of the VTAM working set increase is due to the 3.9% increase in the number of users that were run. However, most of the increase comes from changing from CTCA to 3745 communications. One contributing factor is that message buffers are held longer in the 3745 case because of the lower bandwidth.

The ITR increase is entirely due to a decrease in VTAM processor usage. This is apparent from the results because the decrease in TOT CPU/CMD (V) for the VTAM machine (-0.59 ms) is very similar to the system-wide decrease in total processor usage (-0.56 ms, PBT/CMD (H)). There are two main reasons for this improvement:

1. Some of the processing that was done by VTAM in the CTCA case is instead handled by the 3745.

2. There is a 72% decrease in DIAGNOSE code X′98′ (DIAG 98/CMD (V)), which VTAM uses to do network I/O. This decrease occurs as a consequence of the longer response times, which allow more time for multiple messages to accumulate and be transmitted as a single DIAGNOSE code X′98′ request.

The large changes in trivial response time (TRIV INT) and total internal response time (TOT INT) are mostly due to a large difference in what CP determines to be a transaction. This is reflected by the 47% increase in the ETR metric, which is the number of CP-determined transactions per second (as reported by RTM).

CP has no way of knowing with certainty when one transaction stops and the next transaction starts, so it does the best it can. Whenever a virtual machine is in voluntary wait for more than 300 milliseconds, CP assumes that a new transaction has started. In the CTCA case, the CTCA is so fast that short interactions that happen to be separated by a short think time (such as 0.1 seconds) are so close together that they are separated by less than this 300 milliseconds. Whenever this happens, two real user/system interactions get combined into one CP transaction. In the 3745 case, these same interactions appear to CP as separate transactions.

The true change in internal response time is represented by TOT INT ADJ, which normalizes TOT INT to ETR (T), the TPNS-based transaction rate. ETR (T) counts the number of actual user/system interactions and is consistent in meaning from one measurement to another.

Table 34 (Page 1 of 3). 3745 comparison to CTCA / FS7FMAXR on the 9121-480

Communications **Release** **Run ID**	**CTCA** **VM/ESA 2.1** **L26S1563**	**3745** **VM/ESA 2.1** **L26S1620**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1560**	**1620**	**60**	**3.85%**
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.110	0.051	-0.059	-53.64%
NONTRIV INT	0.421	0.444	0.023	5.46%
TOT INT	0.319	0.246	-0.073	-22.88%
TOT INT ADJ	0.280	0.312	0.032	11.43%
AVG FIRST (T)	0.228	0.274	0.046	20.18%
AVG LAST (T)	0.346	0.482	0.136	39.51%
Throughput				
AVG THINK (T)	26.25	26.49	0.24	0.91%
ETR	48.21	70.93	22.72	47.13%
ETR (T)	54.92	55.92	1.00	1.82%
ETR RATIO	0.878	1.268	0.391	44.50%
ITR (H)	62.58	63.69	1.11	1.78%
ITR	27.49	40.41	12.93	47.03%
EMUL ITR	41.17	60.32	19.15	46.53%
ITRR (H)	1.000	1.018	0.018	1.78%
ITRR	1.000	1.470	0.470	47.03%

Table 34 (Page 2 of 3). 3745 comparison to CTCA / FS7FMAXR on the 9121-480				
Communications **Release** **Run ID**	**CTCA** **VM/ESA 2.1** **L26S1563**	**3745** **VM/ESA 2.1** **L26S1620**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1560**	**1620**	**60**	**3.85%**
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Proc. Usage				
PBT/CMD (H)	31.961	31.403	-0.558	-1.74%
PBT/CMD	31.865	31.474	-0.391	-1.23%
CP/CMD (H)	11.251	10.972	-0.279	-2.48%
CP/CMD	10.561	10.372	-0.189	-1.79%
EMUL/CMD (H)	20.710	20.431	-0.278	-1.34%
EMUL/CMD	21.304	21.102	-0.202	-0.95%
Processor Util.				
TOTAL (H)	175.52	175.60	0.08	0.04%
TOTAL	175.00	176.00	1.00	0.57%
UTIL/PROC (H)	87.76	87.80	0.04	0.04%
UTIL/PROC	87.50	88.00	0.50	0.57%
TOTAL EMUL (H)	113.74	114.25	0.51	0.45%
TOTAL EMUL	117.00	118.00	1.00	0.85%
MASTER TOTAL (H)	87.59	87.85	0.26	0.30%
MASTER TOTAL	88.00	88.00	0.00	0.00%
MASTER EMUL (H)	52.00	52.32	0.32	0.61%
MASTER EMUL	54.00	54.00	0.00	0.00%
TVR(H)	1.54	1.54	-0.01	-0.41%
TVR	1.50	1.49	0.00	-0.28%
Storage				
NUCLEUS SIZE (V)	2364KB	2364KB	0KB	0.00%
TRACE TABLE (V)	400KB	400KB	0KB	0.00%
WKSET (V)	79	79	0	0.00%
PGBLPGS	48821	48537	-284	-0.58%
PGBLPGS/USER	31.3	30.0	-1.3	-4.26%
FREEPGS	4262	4495	233	5.47%
FREE UTIL	0.91	0.90	-0.01	-1.60%
SHRPGS	1259	1239	-20	-1.59%
Paging				
READS/SEC	516	522	6	1.16%
WRITES/SEC	342	347	5	1.46%
PAGE/CMD	15.623	15.540	-0.083	-0.53%
PAGE IO RATE (V)	126.000	127.200	1.200	0.95%
PAGE IO/CMD (V)	2.294	2.275	-0.020	-0.85%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.701	6.778	0.077	1.15%
Queues				
DISPATCH LIST	32.35	39.51	7.16	22.13%
ELIGIBLE LIST	0.00	0.00	0.00	na

Table 34 (Page 3 of 3). 3745 comparison to CTCA / FS7FMAXR on the 9121-480				
Communications **Release** **Run ID**	**CTCA** **VM/ESA 2.1** **L26S1563**	**3745** **VM/ESA 2.1** **L26S1620**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1560**	**1620**	**60**	**3.85%**
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
I/O				
VIO RATE	528	484	-44	-8.33%
VIO/CMD	9.614	8.655	-0.959	-9.97%
RIO RATE (V)	347	297	-50	-14.41%
RIO/CMD (V)	6.318	5.311	-1.007	-15.94%
MDC READS	261	268	7	2.68%
MDC WRITES	123	129	6	4.88%
MDC MODS	77	80	3	3.90%
MDC HIT RATIO	0.84	0.84	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	24.777	24.847	0.070	0.28%
DIAG/CMD	24.411	23.371	-1.039	-4.26%
DIAG 08/CMD	0.759	0.760	0.001	0.13%
DIAG 14/CMD	0.024	0.025	0.000	0.51%
DIAG 58/CMD	1.251	1.250	-0.001	-0.06%
DIAG 98/CMD	1.478	0.418	-1.060	-71.72%
DIAG A4/CMD	2.095	2.101	0.005	0.25%
DIAG A8/CMD	1.631	1.648	0.017	1.02%
DIAG 214/CMD	12.240	12.213	-0.027	-0.22%
SIE/CMD	62.584	58.031	-4.553	-7.27%
SIE INTCPT/CMD	45.060	41.202	-3.858	-8.56%
FREE TOTL/CMD	72.416	71.515	-0.902	-1.25%
VTAM Machines				
WKSET (V)	461	631	170	36.88%
TOT CPU/CMD (V)	4.1172	3.5269	-0.5903	-14.34%
CP CPU/CMD (V)	1.7399	1.4903	-0.2496	-14.35%
VIRT CPU/CMD (V)	2.3773	2.0367	-0.3406	-14.33%
DIAG 98/CMD (V)	1.477	0.418	-1.059	-71.71%
SFS Servers				
WKSET (V)	2997	3204	207	6.91%
TOT CPU/CMD (V)	3.6620	3.7256	0.0636	1.74%
CP CPU/CMD (V)	1.7096	1.7486	0.0390	2.28%
VIRT CPU/CMD (V)	1.9524	1.9771	0.0247	1.27%
FP REQ/CMD(Q)	1.263	1.269	0.006	0.48%
IO/CMD (Q)	1.821	1.891	0.070	3.84%
IO TIME/CMD (Q)	0.022	0.024	0.002	9.09%
SFS TIME/CMD (Q)	0.029	0.032	0.003	10.34%
Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Q=Query Filepool Counters, Unmarked=RTM				

CP Monitor Overhead

Workload: FS7F0R

Hardware Configuration

Processor model:	9121-480
Processors used:	2
Storage	
Real:	224MB
Expanded:	32MB (all for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-2	4	16	6	6			
3390-2	3990-3	2						2 R
3390-2	3990-3	4		2	2	16 R		

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	na	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
SMART	1	RTM	16MB/370	3%		QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XC	10000	512	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	1860	User	3MB/XC	100		

Measurement Discussion: The following table shows the results of measurements with and without the CP monitor running. When monitor was running, all sample domains were enabled with the default interval and rate settings (1 minute and 2 seconds respectively). When running monitor, there was an increase of 2.8% in response time and a decrease of 1.2% in ITR (H). The 3.7% increase in CP/CMD (H) (mostly charged to the system) makes up the majority of the ITR difference. There is also a slight increase due to a user running the MONWRITE utility.

In addition to processor costs, there may also be costs in terms of storage and DASD. The *VM/ESA Release 2.1 Performance* manual contains a formula to compute required size of the monitor DCSS. This same formula could be used to estimate the amount of data produced by the MONWRITE utility for each sample interval. A rough rule of thumb for DASD requirements when collecting sample data domains is:

$$total\_bytes = (users \times 500 + devices \times 450) \times samples$$

For the measurements shown, the cost of running monitor is low. However, there are many factors that affect the cost of monitor. Some of these factors are as follows:

Domains and types of data enabled. Sample data is typically less expensive to collect than event data. The measurements in the table below were made without any event data enabled (except monitor domain, which is always enabled). Two additional measurements (not shown) were made on a 9121-742 with an office workload. One measurement was made with the monitor enabled for all sample domains. The other measurement had sample data enabled for all domains and event data for the user domain. Enabling the user event data increased processor utilization by approximately 2%.

Sample interval and rate. The monitor interval is the period of time between the writing of the sample data to the monitor DCSS. A larger interval lowers the cost in terms of external storage and processor resources. The monitor rate is the time between the collection of high frequency sample data (system, user, processor, and I/O domains). The high frequency counters are updated according to the monitor rate, but the data is written on the regular monitor sample interval. Therefore, a larger monitor rate lowers the processor costs, but not the amount of data written.

Processor speed. For sample data, monitor processing by default occurs every minute. A faster processor has a higher transaction rate. Therefore, the monitor costs per command are smaller on a faster processor.

Number and type of users. For user domain sample data, the absolute cost is proportional to the number of users being monitored, while the per-command cost decreases as the users become more active. For user domain event data, the absolute cost is proportional to the activity of the users being monitored, while the per-command cost is a constant.

Number of devices. The monitoring of devices in the I/O domain is analogous to the user domain. The more devices monitored, the greater the costs. If the seeks domain is enabled, it can greatly increase the cost of monitoring.

Master processor considerations. Master processor constraints should be considered because most monitor functions run on the master processor.

Level of VM/ESA. Enhancements were made in VM/ESA Release 2 to improve the performance of user high frequency sampling. See the *VM/ESA Release 2 Performance Report* or the *VM/ESA Release 2.1 Performance* manual for additional information on this enhancement.

Table 35 (Page 1 of 2). Cost of collecting VM/ESA Release 2.1 monitor sample data for CMS minidisk workload on 9121-480.

Monitor **Release** **Run ID**	**Off** **VM/ESA 2.1** **L26E1864**	**On** **VM/ESA 2.1** **L26E1863**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.120	0.126	0.006	5.00%
NONTRIV INT	0.453	0.480	0.027	5.96%
TOT INT	0.342	0.363	0.021	6.14%
TOT INT ADJ	0.302	0.320	0.018	5.90%
AVG FIRST (T)	0.250	0.253	0.002	0.80%
AVG LAST (T)	0.382	0.393	0.011	2.75%
Throughput				
AVG THINK (T)	26.33	26.40	0.07	0.27%
ETR	57.71	57.60	-0.11	-0.19%
ETR (T)	65.28	65.30	0.02	0.04%
ETR RATIO	0.884	0.882	-0.002	-0.23%
ITR (H)	73.45	72.57	-0.88	-1.20%
ITR	32.48	32.01	-0.47	-1.43%
EMUL ITR	47.04	46.94	-0.09	-0.20%
ITRR (H)	1.000	0.988	-0.012	-1.20%
ITRR	1.000	0.986	-0.014	-1.43%
Proc. Usage				
PBT/CMD (H)	27.229	27.559	0.331	1.21%
PBT/CMD	27.268	27.565	0.297	1.09%
CP/CMD (H)	9.139	9.475	0.336	3.67%
CP/CMD	8.426	8.729	0.303	3.60%
EMUL/CMD (H)	18.090	18.085	-0.005	-0.03%
EMUL/CMD	18.843	18.836	-0.007	-0.04%
Processor Util.				
TOTAL (H)	177.74	179.96	2.22	1.25%
TOTAL	178.00	180.00	2.00	1.12%
UTIL/PROC (H)	88.87	89.98	1.11	1.25%
UTIL/PROC	89.00	90.00	1.00	1.12%
TOTAL EMUL (H)	118.09	118.09	0.01	0.01%
TOTAL EMUL	123.00	123.00	0.00	0.00%
MASTER TOTAL (H)	88.44	89.72	1.29	1.45%
MASTER TOTAL	88.00	90.00	2.00	2.27%
MASTER EMUL (H)	53.81	53.13	-0.68	-1.27%
MASTER EMUL	56.00	55.00	-1.00	-1.79%
TVR(H)	1.51	1.52	0.02	1.24%
TVR	1.45	1.46	0.02	1.12%
Storage				
NUCLEUS SIZE (V)	na	2352KB		
TRACE TABLE (V)	na	400KB		
WKSET (V)	na	83		
PGBLPGS	47637	47529	-108	-0.23%
PGBLPGS/USER	25.6	25.6	-0.1	-0.23%
FREEPGS	5127	5202	75	1.46%
FREE UTIL	0.95	0.94	-0.01	-1.44%
SHRPGS	1071	1126	55	5.14%

Table 35 (Page 2 of 2). Cost of collecting VM/ESA Release 2.1 monitor sample data for CMS minidisk workload on 9121-480.

Monitor **Release** **Run ID**	**Off** **VM/ESA 2.1** **L26E1864**	**On** **VM/ESA 2.1** **L26E1863**	**Difference**	**% Difference**
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Paging				
READS/SEC	627	624	-3	-0.48%
WRITES/SEC	423	424	1	0.24%
PAGE/CMD	16.085	16.049	-0.036	-0.23%
PAGE IO RATE (V)	na	172.700		
PAGE IO/CMD (V)	na	2.645		
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.449	6.493	0.044	0.68%
Queues				
DISPATCH LIST	38.23	40.20	1.97	5.15%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	626	625	-1	-0.16%
VIO/CMD	9.590	9.571	-0.019	-0.20%
RIO RATE (V)	na	390		
RIO/CMD (V)	na	5.972		
MDC READS	361	361	0	0.00%
MDC WRITES	179	179	0	0.00%
MDC MODS	145	146	1	0.69%
MDC HIT RATIO	0.91	0.91	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	17.560	17.577	0.018	0.10%
DIAG/CMD	26.897	26.863	-0.034	-0.13%
DIAG 08/CMD	0.760	0.761	0.001	0.16%
DIAG 14/CMD	0.025	0.025	0.000	0.11%
DIAG 58/CMD	1.250	1.250	0.000	-0.01%
DIAG 98/CMD	1.231	1.187	-0.044	-3.54%
DIAG A4/CMD	3.908	3.914	0.006	0.15%
DIAG A8/CMD	1.882	1.900	0.018	0.96%
DIAG 214/CMD	12.837	12.847	0.010	0.08%
SIE/CMD	54.873	54.777	-0.096	-0.17%
SIE INTCPT/CMD	37.863	37.796	-0.066	-0.17%
FREE TOTL/CMD	62.885	64.532	1.647	2.62%
VTAM Machines				
WKSET (V)	na	519		
TOT CPU/CMD (V)	na	3.8987		
CP CPU/CMD (V)	na	1.6599		
VIRT CPU/CMD (V)	na	2.2388		
DIAG 98/CMD (V)	na	1.185		

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

Minidisk Cache Tuning: Restricting the Arbiter

The CP arbiter function dynamically determines how much expanded storage to use for paging and how much to use for the minidisk cache. Experience has shown that there are cases where overall system performance can be improved by placing restrictions on the arbiter. One case that illustrates this is with the FS7F0R CMS-intensive workload, when run on a 9121-742 processor with 1GB of central storage and 1GB of expanded storage. In this case, it was found that performance improved when the RETAIN command was used to set a 64MB floor for the minidisk cache size. This section summarizes those results and mentions other cases where constraining the arbiter in some way may be of value.

Workload: FS7F0R

Hardware Configuration

Processor model:	9121-742
Processors used:	4
Storage	
Real:	1024MB
Expanded:	1024MB
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-2	2	6	5				
3390-2	3990-2	4	16	6	6			
3390-2	3990-3	4		2	7	32 R		2 R

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
SMART	1	RTM	16MB/370	3%	500	QUICKDSP ON
VSCSn	3	VSCS	64MB/XA	10000	1200	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	550	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
Users	5400	User	3MB/XC	100		

Measurement Discussion: A measurement was obtained where the CP arbiter was unrestricted. This was compared to a matching measurement where RETAIN XSTOR MDC 64M was used to prevent the minidisk cache size from dropping below 64MB.

The results show that the system performed better with the 64MB floor. External response time (AVG LAST (T)) decreased by 7.7% while the internal throughput rate (ITR (H)) increased by 1.0%.

In the unrestricted case, the average size of the minidisk cache was 10.8MB (1.1% of all expanded storage) and this resulted in a MDC hit ratio of 0.74. When a 64MB floor was specified, the average size of the minidisk cache was 66.8MB (6.5% of all expanded storage), yielding a hit ratio of 0.91. This increase in MDC size reduced minidisk I/Os (NONPAGE RIO/CMD (V)) by 0.58 per command, while paging was essentially unchanged.

There are two primary reasons for the ITR improvement. First, there was less I/O processing in CP because there were 0.58 fewer I/Os per command. Second, CP processing was saved because the 64MB floor resulted in fewer changes to the minidisk cache size. As shown by the MINIDISK_CACHE_BY_TIME VMPRF report, the cache size stayed at or near the 64MB floor for extended periods of time.

These results illustrate that although the arbiter usually does a reasonable job of optimizing overall system performance, there are occasions where placing some restrictions on the arbiter can yield significantly better results. Here are some suggestions:

1. If the MDC hit ratio is rather low (below 80%, for example) and yet only a small percentage of expanded storage is being used for MDC, try setting a floor. As was done in the situation shown here, this floor should be set significantly higher than the current average cache size. You might try setting the floor based on the following guideline: MDC size = 10MB + (logged users/128)MB.

2. Conversely, if the MDC hit ratio is high (above 90%), a large percentage of expanded storage is being used for MDC (and perhaps exceeds the above guideline), and the expanded storage paging rates are high, try setting a ceiling. This is done by specifying RETAIN XSTOR MDC 0M xxxM.

3. If you know that the system′s MDC requirements are fairly constant, you can eliminate the processing associated with dynamic changes to the minidisk cache size by fixing the MDC size. Do this by specifying RETAIN XSTOR MDC xxxM xxxM, where the minimum and maximum are set to the same value.

4. If the processor′s storage can be configured as either central or expanded storage, it is often a good strategy to configure most of it as central storage, with the remaining storage configured as expanded storage for use by minidisk caching. The 9121-480 measurements in this report are an example of this situation. If you do that, specify RETAIN XSTOR MDC ALL to ensure that it is all used for MDC. If you specify RETAIN XSTOR MDC xxxM where xxxM is equal to the total expanded storage size, there will still be a small amount of (unproductive) paging to expanded storage.

A given system will often have quite different paging and MDC requirements over time. This means that in some cases you may wish to run with a different RETAIN XSTOR setting at different times of the day. In addition, this is an important reason for leaving the arbiter with as much flexibility as you can.

In the case described in this section, a 64MB floor still meant that only 6.5% of all expanded storage was being committed to MDC. This is important because it means that even in the worst case (there are times when the system has very high paging requirements and no need for MDC), this restraint on the arbiter will still have very little adverse effect on system performance. When a floor or ceiling takes away only a small percentage of total expanded storage from the arbiter′s scope of discretion, this reduced flexibility can do little harm and, as illustrated by the results presented here, can be of significant benefit.

Table 36 (Page 1 of 3). Minidisk cache tuning: The effect of setting a 64MB floor

Minimum MDC Size **Release** **Run ID**	**0MB** **VM/ESA 2.1** **S46E5401**	**64MB** **VM/ESA 2.1** **S46E5400**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**5400**	**5400**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
Response Time				
TRIV INT	0.094	0.097	0.003	3.19%
NONTRIV INT	0.374	0.362	-0.012	-3.21%
TOT INT	0.257	0.254	-0.003	-1.17%
TOT INT ADJ	0.262	0.253	-0.009	-3.61%
AVG FIRST (T)	0.304	0.283	-0.022	-7.12%
AVG LAST (T)	0.413	0.381	-0.032	-7.74%
Throughput				
AVG THINK (T)	26.30	26.26	-0.04	-0.14%
ETR	193.18	188.50	-4.68	-2.42%
ETR (T)	189.20	189.31	0.10	0.05%
ETR RATIO	1.021	0.996	-0.025	-2.48%
ITR (H)	212.21	214.25	2.05	0.97%
ITR	54.20	53.40	-0.80	-1.47%
EMUL ITR	82.10	80.52	-1.58	-1.93%
ITRR (H)	1.000	1.010	0.010	0.97%
ITRR	1.000	0.985	-0.015	-1.47%

Table 36 (Page 2 of 3). Minidisk cache tuning: The effect of setting a 64MB floor				
Minimum MDC Size **Release** **Run ID**	**0MB** **VM/ESA 2.1** **S46E5401**	**64MB** **VM/ESA 2.1** **S46E5400**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**5400**	**5400**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
Proc. Usage				
PBT/CMD (H)	18.850	18.669	-0.180	-0.96%
PBT/CMD	18.869	18.647	-0.222	-1.17%
CP/CMD (H)	6.802	6.673	-0.129	-1.90%
CP/CMD	6.448	6.286	-0.162	-2.51%
EMUL/CMD (H)	12.047	11.996	-0.051	-0.42%
EMUL/CMD	12.421	12.361	-0.060	-0.48%
Processor Util.				
TOTAL (H)	356.64	353.42	-3.22	-0.90%
TOTAL	357.00	353.00	-4.00	-1.12%
UTIL/PROC (H)	89.16	88.36	-0.80	-0.90%
UTIL/PROC	89.25	88.25	-1.00	-1.12%
TOTAL EMUL (H)	227.94	227.10	-0.84	-0.37%
TOTAL EMUL	235.00	234.00	-1.00	-0.43%
MASTER TOTAL (H)	91.37	90.48	-0.89	-0.98%
MASTER TOTAL	91.00	90.00	-1.00	-1.10%
MASTER EMUL (H)	41.60	41.15	-0.45	-1.08%
MASTER EMUL	43.00	43.00	0.00	0.00%
TVR(H)	1.56	1.56	-0.01	-0.54%
TVR	1.52	1.51	-0.01	-0.70%
Storage				
NUCLEUS SIZE (V)	2368KB	2368KB	0KB	0.00%
TRACE TABLE (V)	800KB	800KB	0KB	0.00%
WKSET (V)	69	69	0	0.00%
PGBLPGS	236 k	235 k	-1 k	-0.42%
PGBLPGS/USER	43.7	43.5	-0.2	-0.42%
FREEPGS	14570	14546	-24	-0.16%
FREE UTIL	0.91	0.92	0.00	0.16%
SHRPGS	1486	1493	7	0.47%
Paging				
READS/SEC	525	542	17	3.24%
WRITES/SEC	312	325	13	4.17%
PAGE/CMD	4.424	4.580	0.156	3.53%
PAGE IO RATE (V)	110.700	117.100	6.400	5.78%
PAGE IO/CMD (V)	0.585	0.619	0.033	5.72%
XSTOR IN/SEC	899	888	-11	-1.22%
XSTOR OUT/SEC	1321	1326	5	0.38%
XSTOR/CMD	11.733	11.695	-0.038	-0.32%
FAST CLR/CMD	6.570	6.529	-0.041	-0.62%
Queues				
DISPATCH LIST	106.29	101.98	-4.31	-4.05%
ELIGIBLE LIST	0.02	0.00	-0.02	-100.00%

Table 36 (Page 3 of 3). Minidisk cache tuning: The effect of setting a 64MB floor

Minimum MDC Size **Release** **Run ID**	**0MB** **VM/ESA 2.1** **S46E5401**	**64MB** **VM/ESA 2.1** **S46E5400**	**Difference**	**% Difference**
Environment				
Real Storage	**1024MB**	**1024MB**		
Exp. Storage	**1024MB**	**1024MB**		
Users	**5400**	**5400**		
VTAMs	**1**	**1**		
VSCSs	**3**	**3**		
Processors	**4**	**4**		
I/O				
VIO RATE	1653	1653	0	0.00%
VIO/CMD	8.737	8.732	-0.005	-0.05%
RIO RATE (V)	679	576	-103	-15.17%
RIO/CMD (V)	3.589	3.043	-0.546	-15.22%
NONPAGE RIO/CMD (V)	3.004	2.424	-0.580	-19.3%
MDC READS	833	1047	214	25.69%
MDC WRITES	693	518	-175	-25.25%
MDC MODS	408	421	13	3.19%
MDC HIT RATIO	0.74	0.91	0.17	22.97%
PRIVOPs				
PRIVOP/CMD	20.523	20.529	0.005	0.03%
DIAG/CMD	24.862	24.845	-0.018	-0.07%
DIAG 08/CMD	0.759	0.760	0.001	0.10%
DIAG 14/CMD	0.025	0.025	0.000	0.20%
DIAG 58/CMD	1.251	1.250	0.000	-0.04%
DIAG 98/CMD	0.357	0.363	0.006	1.66%
DIAG A4/CMD	3.905	3.901	-0.004	-0.11%
DIAG A8/CMD	1.888	1.889	0.001	0.05%
DIAG 214/CMD	12.978	12.964	-0.014	-0.11%
SIE/CMD	52.854	52.539	-0.314	-0.59%
SIE INTCPT/CMD	35.940	35.727	-0.214	-0.59%
FREE TOTL/CMD	63.424	63.389	-0.035	-0.05%
VTAM Machines				
WKSET (V)	4123	4108	-15	-0.36%
TOT CPU/CMD (V)	2.6544	2.6471	-0.0073	-0.28%
CP CPU/CMD (V)	1.2802	1.2737	-0.0065	-0.51%
VIRT CPU/CMD (V)	1.3742	1.3734	-0.0008	-0.06%
DIAG 98/CMD (V)	0.357	0.363	0.006	1.74%

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

High Level Assembler Evaluation

Workload: FS7F0R

Hardware Configuration

Processor model:	9121-480
Processors used:	2
Storage	
Real:	224MB
Expanded:	32MB (all for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-3	2						2 R
3390-2	3990-2	4	16	4	6			
3390-2	3990-3	4		3	2	16 R		

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
SMART	1	RTM	16MB/370	3%	400	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	512	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	100		QUICKDSP ON
User	1860	Users	3MB/XC	100		

Measurement Discussion: The following table summarizes the results of comparing the Assembler H Version 2 (HASM) to the IBM High Level Assembler (HLASM) when running on VM/ESA Release 2.1. The assembler commands within the FS7F workload represent less than 2% of the total system load. The FS7F workload uses the assembler options of NOLIST and OBJECT. The major difference in options is that HLASM used the DECK option and HASM used the NODECK option. It should be noted that the FS7F workload assembles only one source file with some macro expansions and does not claim this to be representative.

The following was extracted from the *IBM High Level Asembler for MVS, VM and VSE Presentation Guide*. See that document for descriptions of the functional enhancements made.

C-17 Key Features: Other Enhancements

6. Performance improvements have been measured in two groups: for assemblies with few or no macros, and for those with heavy macro usage:

   - Few or no macros

     High Level Assembler provides performance benefits in all areas.

   - Heavy macro usage

     CMS and MVS show that additional processor resource and elapsed time are required. The reason for this is that High Level Assembler provides a much richer set of services and information for macros (for example, the 19 new system variable symbols); more processor time is needed to support them. Because machine resources are becoming cheaper at the same time that human resources with Assembler Language skills are becoming more expensive, it is a design factor for High Level Assembler to "trade machine time for people time." Thus, we expect that the added costs of machine time for assembling certain macro-based applications will be more than repaid by the savings in the time and effort needed to support them.

   Substantial improvements are provided in the VSE/ESA environment, in all cases.

The 9121-480 experienced a 0.6% decrease in internal throughput rate (ITR (H)) and a 5.7% improvement in external response time (AVG LAST (T)).

Assemble command data (not shown in the following table) were extracted from TPNS reports. External response time to do the assembly improved by 1.8%, while total processor time increased by 34.4%.

Table 37 (Page 1 of 2). Assembler comparison on VM/ESA Release 2.1 on the 9121-480

Assembler Release Run ID	HASM VM/ESA 2.1 L26E1862	HLASM VM/ESA 2.1 L26E1860	Difference	% Difference
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Response Time				
TRIV INT	0.135	0.124	-0.011	-8.15%
NONTRIV INT	0.510	0.480	-0.030	-5.88%
TOT INT	0.383	0.361	-0.022	-5.74%
TOT INT ADJ	0.341	0.321	-0.020	-5.83%
AVG FIRST (T)	0.281	0.266	-0.015	-5.17%
AVG LAST (T)	0.433	0.408	-0.025	-5.66%
Throughput				
AVG THINK (T)	26.33	26.36	0.03	0.11%
ETR	58.16	58.14	-0.02	-0.03%
ETR (T)	65.38	65.42	0.04	0.06%
ETR RATIO	0.890	0.889	-0.001	-0.10%
ITR (H)	72.72	72.31	-0.42	-0.57%
ITR	32.34	32.13	-0.21	-0.65%
EMUL ITR	47.30	46.81	-0.49	-1.03%
ITRR (H)	1.000	0.994	-0.006	-0.57%
ITRR	1.000	0.994	-0.006	-0.65%
Proc. Usage				
PBT/CMD (H)	27.501	27.660	0.159	0.58%
PBT/CMD	27.530	27.666	0.136	0.49%
CP/CMD (H)	9.412	9.381	-0.031	-0.33%
CP/CMD	8.718	8.712	-0.005	-0.06%
EMUL/CMD (H)	18.089	18.278	0.190	1.05%
EMUL/CMD	18.812	18.953	0.141	0.75%
Processor Util.				
TOTAL (H)	179.81	180.96	1.15	0.64%
TOTAL	180.00	181.00	1.00	0.56%
UTIL/PROC (H)	89.91	90.48	0.57	0.64%
UTIL/PROC	90.00	90.50	0.50	0.56%
TOTAL EMUL (H)	118.27	119.59	1.31	1.11%
TOTAL EMUL	123.00	124.00	1.00	0.81%
MASTER TOTAL (H)	89.62	90.07	0.45	0.50%
MASTER TOTAL	90.00	90.00	0.00	0.00%
MASTER EMUL (H)	53.13	53.65	0.52	0.97%
MASTER EMUL	55.00	56.00	1.00	1.82%
TVR(H)	1.52	1.51	-0.01	-0.47%
TVR	1.46	1.46	0.00	-0.26%
Storage				
NUCLEUS SIZE (V)	2348KB	2348KB	0KB	0.00%
TRACE TABLE (V)	400KB	400KB	0KB	0.00%
WKSET (V)	82	82	0	0.00%
PGBLPGS	47571	47545	-26	-0.05%
PGBLPGS/USER	25.6	25.6	0.0	-0.05%
FREEPGS	5162	5184	22	0.43%
FREE UTIL	0.94	0.94	0.00	-0.42%
SHRPGS	1135	1121	-14	-1.23%

Table 37 (Page 2 of 2). Assembler comparison on VM/ESA Release 2.1 on the 9121-480

Assembler Release Run ID	HASM VM/ESA 2.1 L26E1862	HLASM VM/ESA 2.1 L26E1860	Difference	% Difference
Environment				
Real Storage	**224MB**	**224MB**		
Exp. Storage	**32MB**	**32MB**		
Users	**1860**	**1860**		
VTAMs	**1**	**1**		
VSCSs	**0**	**0**		
Processors	**2**	**2**		
Paging				
READS/SEC	625	624	-1	-0.16%
WRITES/SEC	424	426	2	0.47%
PAGE/CMD	16.044	16.049	0.005	0.03%
PAGE IO RATE (V)	173.900	173.800	-0.100	-0.06%
PAGE IO/CMD (V)	2.660	2.657	-0.003	-0.12%
XSTOR IN/SEC	0	0	0	na
XSTOR OUT/SEC	0	0	0	na
XSTOR/CMD	0.000	0.000	0.000	na
FAST CLR/CMD	6.454	6.527	0.072	1.12%
Queues				
DISPATCH LIST	41.77	37.95	-3.82	-9.15%
ELIGIBLE LIST	0.00	0.00	0.00	na
I/O				
VIO RATE	622	620	-2	-0.32%
VIO/CMD	9.513	9.477	-0.036	-0.38%
RIO RATE (V)	392	389	-3	-0.77%
RIO/CMD (V)	5.995	5.946	-0.050	-0.83%
MDC READS	361	368	7	1.94%
MDC WRITES	178	172	-6	-3.37%
MDC MODS	145	139	-6	-4.14%
MDC HIT RATIO	0.91	0.91	0.00	0.00%
PRIVOPs				
PRIVOP/CMD	17.718	17.630	-0.088	-0.50%
DIAG/CMD	26.774	26.652	-0.122	-0.46%
DIAG 08/CMD	0.762	0.761	0.000	-0.04%
DIAG 14/CMD	0.025	0.025	0.000	-0.14%
DIAG 58/CMD	1.243	1.242	-0.001	-0.06%
DIAG 98/CMD	1.141	1.175	0.034	2.99%
DIAG A4/CMD	3.906	3.852	-0.054	-1.37%
DIAG A8/CMD	1.896	1.890	-0.006	-0.32%
DIAG 214/CMD	12.830	12.744	-0.086	-0.67%
SIE/CMD	54.509	54.720	0.211	0.39%
SIE INTCPT/CMD	37.611	37.757	0.146	0.39%
FREE TOTL/CMD	64.465	64.242	-0.223	-0.35%
VTAM Machines				
WKSET (V)	525	521	-4	-0.76%
TOT CPU/CMD (V)	3.8660	3.8467	-0.0193	-0.50%
CP CPU/CMD (V)	1.6314	1.6219	-0.0095	-0.58%
VIRT CPU/CMD (V)	2.2347	2.2248	-0.0099	-0.44%
DIAG 98/CMD (V)	1.141	1.175	0.034	2.96%

Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM

VM/ESA REXX Performance Guidelines

REXX performance suggestions are consolidated and quantified in this section. The majority of REXX applications may not need to be scrutinized with respect to performance, but a performance-sensitive REXX application may be optimized by selecting the best REXX environment and using a coding style that reduces the cost of REXX scanning and executing. Execs, such as those on shared disks, that are commonly run by many users on a system are good candidates for performance tuning.

Although not quantified here, a REXX application's performance may be improved by loading the REXX program into memory with EXECLOAD to save the cost of loading the exec in future executions or saving the exec in a shared segment to reduce overall system virtual storage requirements. Also, if the REXX compiler is installed on the system, the REXX program may be compiled to reduce run time. These suggestions and most of those that are provided later in this section apply to CMS Pipelines written in REXX as well.

REXX Environments

The system under which REXX programs run includes a default environment for processing commands. Selecting the optimum REXX environment within your REXX program will reduce the amount of searching required for issuing commands.

When it begins scanning and executing a file, the REXX interpreter establishes, by default, the XEDIT environment for XEDIT macros and the CMS environment for CMS execs. For example, if you issued the CP command SP READER NOHOLD from an XEDIT macro, the default search path would be as follows:

1. The REXX interpreter determines if it is a REXX instruction. If not, it passes it to XEDIT.
2. XEDIT determines if it is a XEDIT instruction. If not, it passes it to CMS.
3. CMS determines if it is a CMS instruction. If not, it passes it to CP.
4. CP runs the instruction.

Within your program, selecting the optimum REXX environment can reduce the amount of searching. (Issuing the command SAY ADDRESS() will tell you which environment is active.) The following are three environments that REXX understands along with the order in which the command search is performed when all defaults apply:

- XEDIT: editor command, exec, CMS command, CP command
- CMS: exec, CMS command, CP command
- COMMAND: CMS command

The REXX environment can be changed temporarily or permanently to reduce the amount of searching. For example, you can temporarily change to the command environment for the ACCESS CMS command by issuing

```
address 'COMMAND' 'ACCESS ...'
```

then for subsequent commands use the current default environment. The REXX environment can be changed permanently by setting a new default, so subsequent commands are sent to the new environment as follows:

```
 address 'COMMAND'   /* sets new default environment */
   .
   .
'ACCESS ...'         /* uses new default environment */
```

"Set Default REXX Environment: address 'COMMAND'" on page 172 describes the performance characteristics both for changing the environment for one command and for setting a new default.

Upon entry to a function or subroutine, the environment of the caller is used and upon return, the environment of the caller is restored.

XEDIT search order: XEDIT goes through the following steps to determine if an instruction is an XEDIT instruction.[8]

1. If invoked as

   ```
   COMMAND subcommand
   ```

   then only try steps 4, 5, 6a, and 7. (This is the editor command COMMAND, not the REXX environment COMMAND.)

2. If invoked as

   ```
   MACRO macro
   ```

   then only try step 6b.

3. Look for a synonym and substitute (unless SET SYNONYM OFF).

4. If command is "=" then rerun last command.

5. Look for an implicit locate (for example ":2" is an implicit form of "LOCATE :2").

6. The default for SET MACRO ON or OFF is OFF. (If SET MACRO ON has been specified, check step 6b before 6a):

   a. Look for an XEDIT subcommand. If none, check for implicit editor SET subcommand. (For example VERIFY ON 1 80 is really SET VERIFY ON 1 80).
   b. Look for an XEDIT macro.

7. If XEDIT SET IMPCMSCP ON, then pass to CMS.

Note: Setting the default to MACRO ON is not functionally equivalent to prefixing with MACRO. This is because fewer areas are searched when looking for the command when it is prefixed with MACRO.

[8] In this discussion, XEDIT macro is an editor command that is implemented in an exec file with a file type of XEDIT. XEDIT subcommands are the other commands that XEDIT understands, but are not in a file with file type XEDIT.

REXX Scanning and Executing

Scanning involves combining the separate characters of the file into higher constructs. For example, the DO construct would be formed from the letter D followed by the letter O during scanning. The scanning virtual path length applies just once per invocation of the REXX program. The execution virtual path length applies each time an instruction is executed. Total virtual path length represents the sum of scanning and execution path length.

REXX scans and executes a program as follows:

- From the top, scan and execute the first instruction, then scan and execute the second, and so on.
- However, to execute a call to a subroutine or a function call, REXX typically
  - Goes directly to the specified label if the label is already known (previously scanned), or
  - Begins scanning (without executing):
    - if the label is found, begin to scan and execute each instruction
    - if the label is not found, look outside the program (the entire program has now been scanned)
- You can tell the interpreter to look outside the program for the function or subroutine, without scanning, by putting quotes around the name. For example:

```
x = QUEUED()    initiates a scan

x = 'QUEUED'()  looks directly for external function
```

  Note: Using quotes suppresses a cache of the label. If the quoted code is in a loop, you may want to try it both ways (see "Quoted Function, Subroutine Trade-off" on page 174).

Summary of REXX Performance Tips

The measurements and conclusions summarized in the following tables were based on

VM/ESA Release 2. Changes in VM/ESA internals could change the results. However, the recommendations should be fairly stable.

virtual machine configuration. For example, the cost of searching all accessed minidisks or directories for an exec varies with the number of accessed minidisks and (to some extent) the number of files on each minidisk.

small test programs. For example, saving a variable lookup could be more of a benefit to programs that contain more variables.

CMS path lengths. The tips assume that CP path lengths would not change the recommendations.

Execution from FILELIST The data supplied were gathered by running the test cases from FILELIST. The path lengths will be different if run by other means, but the relative positioning of the various cases will be the same.

System defaults were used for the measurements unless otherwise specified. For example, when measuring the path length of an XEDIT macro, the XEDIT environment was used. (This is the default REXX environment for XEDIT macros; see "REXX Environments" on page 155.)

Alternative REXX coding techniques are compared in this section. The method having the smallest total path length is considered the best and is flagged with a √ on the right of the table entry. Sometimes a second method is worth consideration. If so, it will also be flagged with the √ . The second method may, for example, have a shorter execution path length which would be a benefit when the code is in a loop.

Alternate methods are presented for three reasons:

- To give an approximation of how much improvement may be gained by going to the better performing method.
- There are sometimes functional differences between alternatives, and the best method may not be what you need. By showing some of the alternatives you still have a choice.
- Sometimes there is a trade off between good coding techniques and coding for performance. Performance suggestions will be made even if they are not good coding technique so you are at least aware of the faster option.

Data were collected using the CP TRACE facility. The collection method is described in "Obtaining the Path Lengths" on page 177. This method can be used to measure alternatives in your environment.

Table 38 (Page 1 of 3). Summary of REXX performance tips

Page	Tips
160	"Invoking Commands From an Exec"
160	"To Invoke a CP Command: address ′COMMAND′ ′CP cpcmd′"
160	"To Invoke a CMS command: address ′COMMAND′ ′cmscmd′"
160	"To Invoke an Exec: address ′COMMAND′ ′EXEC exec′"
161	"Additional CMS Short Cuts"
161	"Abbreviate CMS Command Parameters (But Not CMS Commands)"
161	"Use Variable Record Format Files (RECFM=V)"
161	"Use the STEM Option of EXECIO"
162	"With EXECIO Loop, Use Single FINIS"
162	"Use MAKEBUF, DROPBUF Only When Needed"
162	"VALUE is Faster Than GLOBALV For One Variable"
162	"Minimize Wildcards"
163	"Invoking Commands From XEDIT Macros"
163	"To Invoke CP Command: ′CP cpcmd′"
163	"To Invoke CMS Command: address ′COMMAND′ ′cmscmd′"
163	"To Invoke an Exec: address ′COMMAND′ ′EXEC exec′"
164	"To Invoke a Macro: ′MACRO macro′"
164	"To Invoke an Editor Subcommand: ′COMMAND cmd′"
164	"Additional Editor Short Cuts"
164	"Use EXTRACT Instead of CMS Stack"

Table 38 (Page 2 of 3). Summary of REXX performance tips	
Page	**Tips**
164	“SET SYNONYM OFF”
165	“Combine Commands, Macros Sent to the Editor”
165	“Use Implicit SET for SET Editor Command”
165	“Abbreviate XEDIT Subcommands and Parameters”
165	“Accept XEDIT Command Defaults”
165	“Minimize Space”
165	“Remove Blank Lines”
165	“Minimize Spaces Before Code on Line”
166	“Do Not Space Before the Continuation Character”
166	“Working With Comments”
166	“Minimize Characters in Comments”
166	“Execution Path Length = 0 for Code Comments”
166	“Minimize the Number of Consecutive Line Comments”
167	“Use Multiline Comments”
167	“Locate Line Comments Before Labels”
167	“Keep Line Comments Out of Hot Paths”
167	“Working With IF and SELECT Statements”
167	“ELSE Path is Faster for IF-THEN-ELSE”
167	“Order of OR Expressions for THEN path Is Irrelevant”
168	“Use SELECT for OR-THEN Path”
168	“Order of AND Expressions for ELSE Path is Irrelevant”
169	“Use SELECT for AND-ELSE Path”
169	“Use Nested IFs In Place of SELECT”
170	“Use IF Flag THEN ... ELSE”
170	“Use ′DO flag′ In Place of ′IF flag THEN′”
171	“SELECT: Put Most Likely WHENs Higher”
171	“Working With Variables”
171	“To Initialize All Compound Variables, aaaa. = 5”
171	“Use PARSE to Initialize Variables”
171	“Use Short Variable Names”
171	“Start Parts of Compound Name Tail With a Digit”
172	“Use PARSE VAR to Process Words in a Variable”
172	“Other Tips”
172	“Set Default REXX Environment: address ′COMMAND′”
172	“Built-in Functions Are Generally More Efficient.”
172	“Remove Loops When Possible”
173	“Combine Statements Where Logic Permits”
173	“Understand Available REXX Capabilities”
173	“...=... Versus ...==...”
173	“Use the Concatenation Operator Only When Necessary”
174	“Minimize Scanning of Unexecuted Code”
174	“Quoted Function, Subroutine Trade-off”
174	“Enclose Nonvariable Data in Quotes”
175	“Reduce Arithmetic Precision (Set by NUMERIC DIGITS)”
175	“Remove the Semicolon at the End of the Line”
175	“Exit a Loop Via LEAVE Instead of SIGNAL”
176	“Use ITERATE to Alter the Flow in a Loop”
176	“Invoke Code: Inline, Routine, Procedure, External Exec”

Table 38 (Page 3 of 3). Summary of REXX performance tips	
Page	**Tips**
176	"Use EXPOSE to Pass Parameter"
177	"Abbreviate CP Commands and Parameters"
177	"Try Alternate Approaches"

In the following tables, column headings are as follows:

TotlP Total virtual path length (scanning plus execution path lengths combined).

ExcuP Execution virtual path length.

ScanP Scanning virtual path length.

Example REXX code examples for which data is supplied.

Invoking Commands From an Exec

The following shows various ways of invoking CP commands, CMS commands, and execs from an exec (with the default REXX environment of CMS).

To Invoke a CP Command: address 'COMMAND' 'CP cpcmd': The fastest method saved 8522 total path length when compared to the slowest.

TotlP	ExcuP	ScanP	Example
13556	13095	461	SPOOL READER NOHOLD
12753	12500	253	'SPOOL READER NOHOLD'
8689	8410	279	'CP SPOOL READER NOHOLD'
9005	8442	563	address 'CMS' 'CP SPOOL READER NOHOLD'
13457	12532	925	address 'CMS' 'SPOOL READER NOHOLD'
5034	4451	583	address 'COMMAND' 'CP SPOOL READER NOHOLD' √
5406	4789	617	CALL 'DIAG' '08','SPOOL READER NOHOLD'
5813	5140	673	address 'COMMAND' 'EXECIO 0 CP (STRING SPOOL READER NOHOLD'

To Invoke a CMS command: address 'COMMAND' 'cmscmd': The fastest method saved 4104 total path length when compared to the slowest.

TotlP	ExcuP	ScanP	Example
8365	8114	251	MAKEBUF
8156	8028	128	'MAKEBUF'
8472	8060	412	address 'CMS' 'MAKEBUF'
4368	3936	432	address 'COMMAND' 'MAKEBUF' √

To Invoke an Exec: address 'COMMAND' 'EXEC exec': The fastest method saved 21475 total path length when compared to the slowest. NULLEXEC EXEC only contained the following two lines:

```
/* */
exit
```

TotlP	ExcuP	ScanP	Example
29344	28704	640	`address 'CMS' 'EXEC NULLEXEC'`
46969	46463	506	`CALL 'NULLEXEC'`
29094	28666	428	`'EXEC NULLEXEC'`
26492	26041	451	`NULLEXEC`
26284	25951	333	`'NULLEXEC'`
26503	25978	525	`address 'CMS' 'NULLEXEC'`
25494	24945	549	`address 'COMMAND' 'EXEC NULLEXEC'` √

Additional CMS Short Cuts

Abbreviate CMS Command Parameters (But Not CMS Commands): Abbreviating the parameter (but not the command) saved 4229 total path length.

TotlP	ExcuP	ScanP	Example
8775	8308	467	`address 'COMMAND' 'QUERY LANGUAGE'`
12860	12418	442	`address 'COMMAND' 'Q LANGUAGE'`
12716	12294	422	`address 'COMMAND' 'Q LANG'`
8631	8184	447	`address 'COMMAND' 'QUERY LANG'` √

Use Variable Record Format Files (RECFM=V): Scan path length for this one statement was reduced by 312 instructions by using RECFM=V. In this section a • represents the space character in the example column.

TotlP	ExcuP	ScanP	Example
904	391	513	`x=1····...····` (total of 80 characters) • this file was RECFM=F LRECL=80
592	391	201	`x=1` √ • this file was RECFM=V

Use the STEM Option of EXECIO: To read in an entire file, a single EXECIO DISKR with STEM option is better than multiple EXECIOs. The stem option requires more storage than holding one line at a time in memory (the VAR option). The STEM option is also available for writing (EXECIO DISKW).

Using this tip saved 37424 total path length:

TotlP	ExcuP	ScanP	Example

64766	62809	1957	`do i = 1 to 5`
			`   address 'COMMAND' 'EXECIO 1 DISKR LINES05 ASSEMBLE B (VAR X'`
			`   end`
			`address 'COMMAND' 'FINIS LINES05 ASSEMBLE B'`

30046	28805	1241	`address 'COMMAND' 'EXECIO * DISKR LINES05 ASSEMBLE B (STEM Y.'` `address 'COMMAND' 'FINIS LINES05 ASSEMBLE B'`
27342	26626	716	`address 'COMMAND' 'EXECIO * DISKR LINES05 ASSEMBLE B (FINIS STEM Y.'` √

With EXECIO Loop, Use Single FINIS: When multiple EXECIOs are done to same file, it is faster to keep the file open and do a single FINIS at end. Using the single FINIS saved 68421 total path length. For both examples, the two lines beginning with "address" were actually one long line in the test files.

TotlP	ExcuP	ScanP	Example

124439	122796	1643	`do i = 1 to 5`
			`address 'COMMAND' 'EXECIO 1 DISKW LINESXX`
			`ASSEMBLE B (FINIS STRING TEXT'`
			`end`

56018	53973	2045	`do i = 1 to 5` √
			`address 'COMMAND' 'EXECIO 1 DISKW LINESXX`
			`ASSEMBLE B (STRING TEXT'`
			`end`
			`address 'COMMAND' 'FINIS LINES05 ASSEMBLE B'`

Use MAKEBUF, DROPBUF Only When Needed: You can see that they each cost about 4500 total path length. If you issue a MAKEBUF, put entries on the stack, always remove all those entries from the stack, and issue a DROPBUF, then you can remove the MAKEBUF/DROPBUF as it serves no useful purpose.

TotlP	ExcuP	ScanP	Example
4496	4065	431	`address 'COMMAND' 'MAKEBUF'`
4486	4055	431	`address 'COMMAND' 'DROPBUF'`

VALUE is Faster Than GLOBALV For One Variable: Using the REXX VALUE function saved 3613 total path length (7160-3547) when putting a global variable, and saved 4547 total path length (7142-2595) when getting a global variable. However, GLOBALV has the advantage when more than one variable is to be saved/retrieved because GLOBALV supports a list of variables.

TotlP	ExcuP	ScanP	Example
7160	6593	567	`address 'COMMAND' 'GLOBALV SELECT TESTGRP PUT GLOBVAR'`
7142	6575	567	`address 'COMMAND' 'GLOBALV SELECT TESTGRP GET GLOBVAR'`
3547	2692	855	`prevvar = 'VALUE'('VALUVAR',valuvar,'GLOBAL TESTGRP') /* PUT */` √
2595	1932	663	`newvar = 'VALUE'('VALUVAR',,'GLOBAL TESTGRP') /* GET */` √

Minimize Wildcards: By specifying the file mode, 19560 total path length was saved because CMS did not have to search all file modes.

TotlP	ExcuP	ScanP	Example
28151	27634	517	`address 'COMMAND' 'LISTFILE WILDCARD EXEC *'`
8591	8069	522	`address 'COMMAND' 'LISTFILE WILDCARD EXEC B1'` √

Invoking Commands From XEDIT Macros

The following shows various ways of invoking CP and CMS commands, execs, XEDIT macros, and editor subcommands from XEDIT macros (with the default REXX environment of XEDIT).

To Invoke CP Command: 'CP cpcmd': The fastest method saved 15337 total path length when compared to the slowest.

TotlP	ExcuP	ScanP	Example
17691	17255	436	`SPOOL READER NOHOLD`
16870	16642	228	`'SPOOL READER NOHOLD'`
2354	2075	279	`'CP SPOOL READER NOHOLD'` √
7296	6733	563	`address 'CMS' 'CP SPOOL READER NOHOLD'`
11708	10805	903	`address 'CMS' 'SPOOL READER NOHOLD'`
3497	2914	583	`address 'COMMAND' 'CP SPOOL READER NOHOLD'`
4008	3391	617	`CALL 'DIAG' '08','SPOOL READER NOHOLD'`
4320	3647	673	`address 'COMMAND' 'EXECIO 0 CP ( STRING SPOOL READER NOHOLD'`

To Invoke CMS Command: address 'COMMAND' 'cmscmd': The fastest method saved 10977 total path length when compared to the slowest.

TotlP	ExcuP	ScanP	Example
13914	13683	231	`MAKEBUF`
13699	13591	108	`'MAKEBUF'`
8950	8812	138	`'CMS MAKEBUF'`
7040	6638	402	`address 'CMS' 'MAKEBUF'`
2937	2505	432	`address 'COMMAND' 'MAKEBUF'` √
8984	8806	178	`'COMMAND CMS MAKEBUF'`

To Invoke an Exec: address 'COMMAND' 'EXEC exec': The fastest method saved 21306 total path length when compared to the slowest. NULLEXEC EXEC only contained the following two lines:

```
/* */
exit
```

TotlP	ExcuP	ScanP	Example
30812	30419	393	`NULLEXEC`
30604	30329	275	`'NULLEXEC'`
26161	25860	301	`'CMS NULLEXEC'`
23833	22964	869	`address 'CMS' 'NULLEXEC'`
22460	21922	538	`address 'COMMAND' 'EXEC NULLEXEC'` √
26292	25708	584	`address 'CMS' 'EXEC NULLEXEC'`
43766	43318	448	`call 'NULLEXEC'`
33265	32893	372	`'EXEC NULLEXEC'`

To Invoke a Macro: ′MACRO macro′: With the default setting of SET MACRO OFF, using the MACRO prefix saved 835 total path length. Similar tests also showed that the MACRO prefix helped, but not by as much, when SET MACRO ON was active.

A functional difference between using the MACRO prefix and not using the prefix is that using the prefix avoids separating of nonalphabetic characters. For example

```
N2
```

sends parameter 2 to routine N; however,

```
MACRO N2
```

invokes routine N2.

TotlP	ExcuP	ScanP	Example
8218	8084	134	′NULLMACR′
7383	7219	164	′MACRO NULLMACR′ √

To Invoke an Editor Subcommand: ′COMMAND cmd′: Both techniques are very similar (total path length). If ′COMMAND EXTRACT /FNAME/′ were in a loop it would have the advantage because its execution path length was slightly less. Also, if SET MACRO ON had been in effect, ′COMMAND EXTRACT /FNAME/′ would have had a definite advantage because it would save searching for an XEDIT macro named EXTRACT. See "Abbreviate XEDIT Subcommands and Parameters" on page 165 for another way to reduce the path length for this same subcommand.

TotlP	ExcuP	ScanP	Example
3429	3258	171	′EXTRACT /FNAME/′
3438	3227	211	′COMMAND EXTRACT /FNAME/′ √

Additional Editor Short Cuts

Use EXTRACT Instead of CMS Stack: This example gets the logical record length of the file being edited. Use of EXTRACT saved 10684 total path length.

TotlP	ExcuP	ScanP	Example
11029	9713	1316	address ′COMMAND′ ′LISTFILE USEEXTR XEDIT B (FIFO DATE NOHEADER′ parse pull p1 p2 p3 p4 lrecln .
3457	3254	203	′COMMAND EXTRACT /LRECL′ √

SET SYNONYM OFF: Each avoidance of synonym checking saved 172 total path length (3256-3084). So, after 15 avoidances (2612/172), the new default began to help.

TotlP	ExcuP	ScanP	Example
3256	3066	190	′SET PF1 111′
2612 3084	2404 2856	208 228	′COMMAND SET SYNONYM OFF′ ′SET PF1 111′ √

Combine Commands, Macros Sent to the Editor: The last 4 commands in the second method averaged a savings of 760 total path length (15596-12558)/4.

TotlP	ExcuP	ScanP	Example	
15596	14359	1237	'COMMAND SET PF1 111' 'COMMAND SET PF1 111' 'COMMAND SET PF1 111' 'COMMAND SET PF1 111' 'COMMAND SET PF1 111'	
12558	11165	1393	'COMMAND SET PF1 111#', 'COMMAND SET PF1 111#', 'COMMAND SET PF1 111#', 'COMMAND SET PF1 111#', 'COMMAND SET PF1 111'	√

Use Implicit SET for SET Editor Command: Each avoidance saved 32 total path length.

TotlP	ExcuP	ScanP	Example	
3260	3101	159	'COMMAND SET PF1 111'	
3228	3051	177	'COMMAND PF1 111'	√

Abbreviate XEDIT Subcommands and Parameters: Saved about 17 total path length per character removed: (3503-3384)/7

TotlP	ExcuP	ScanP	Example	
3503	3305	198	'COMMAND EXTRACT /FNAME/'	
3452	3269	183	'COMMAND EXTRACT /FN/'	
3384	3221	163	'COMMAND EXT /FN/'	√

Accept XEDIT Command Defaults: Removing the 1 saved 65 total path length.

TotlP	ExcuP	ScanP	Example	
1817	1678	139	'COMMAND N 1'	
1752	1623	129	'COMMAND N'	√

Minimize Space

In these sections a • represents the space character in the example column.

Remove Blank Lines: Removing blank lines saved execution path length=37 and scan path length=59

TotlP	ExcuP	ScanP	Example	
96	37	59	*blank line*	
0	0	0	*no line*	√

Minimize Spaces Before Code on Line: Moving code to the left saved 3 scan path length per character: (216-201)/5.

TotlP	ExcuP	ScanP	Example
663	447	216	`·····x=1`
648	447	201	`x=1` √

Do Not Space Before the Continuation Character: Saved 3 scan path length per character (282-267)/5.

TotlP	ExcuP	ScanP	Example
729	447	282	`x=·····,` `1`
714	447	267	`x=,` `1` √

Working With Comments

In this discussion "line comments" refers to lines that have just a comment (no executable code):

```
/* this is a line comment */
```

"Code comments" refers to comments that are along side the code:

```
x = 1  /* this is a code comment */
```

"Comments" refers to line comments and code comments

Minimize Characters in Comments: Each character removed from a comment saves 5 scan path length (118-103)/3.

TotlP	ExcuP	ScanP	Example
250	147	103	`/**/` √
255	147	108	`/*.*/`
260	147	113	`/*..*/`
265	147	118	`/*...*/`

Execution Path Length = 0 for Code Comments: Code comments do not cause any additional execution path length (they do, however, cause additional scan path length).

TotlP	ExcuP	ScanP	Example
722	447	275	`x=1 /*.....*/`
648	447	201	`x=1` √

Minimize the Number of Consecutive Line Comments: Each additional line comment cost 160 total path length ((590-270)/2) and 37 execution path length ((221-147)/2).

TotlP	ExcuP	ScanP	Example	
270	147	123	`/*....*/`	√
430	184	246	`/*....*/` `/*....*/`	
590	221	369	`/*....*/` `/*....*/` `/*....*/`	

Use Multiline Comments: Switching from three consecutive line comments to a single multiline comment (with three lines) saved 196 total path length (590-394). Also, notice that execution path length for the multiline comment (147) is the same as for a single line comment.

TotlP	ExcuP	ScanP	Example	
270	147	123	`/*....*/`	
590	221	369	`/*....*/` `/*....*/` `/*....*/`	

394	147	247	`/*....`
			`  ....`
			`  ....*/`
				√

Locate Line Comments Before Labels: It costs 37 execution path length to execute a line comment (see "Minimize the Number of Consecutive Line Comments" on page 166).

Keep Line Comments Out of Hot Paths: It costs 37 execution path length to execute a line comment (see "Minimize the Number of Consecutive Line Comments" on page 166).

Working With IF and SELECT Statements

ELSE Path is Faster for IF-THEN-ELSE

TotlP	ExcuP	ScanP	Example	

2804
2766
	1386
	1348
		1418
		1418
			(with g=0, THEN path taken)
			(with g=1, ELSE path taken)
			
			`if go=0`
			`   then do`
			`     end`
			`   else do`
			`     end`
				
				√

Order of OR Expressions for THEN path Is Irrelevant: It is not faster to put the most frequent true condition first because all three conditions are evaluated anyway. However, this may influence the way you code (see "Use SELECT for OR-THEN Path" on page 168).

TotlP	ExcuP	ScanP	Example

4529
4529
4532
	2709
	2709
	2712
		1820
		1820
		1820
			(when a=1 b=0 c=0)
			(when a=0 b=1 c=0)
			(when a=0 b=0 c=1)
			
			if a=1 | b=1 | c=1
			   then do
			      end
			   else do
			      end

Use SELECT for OR-THEN Path: From section "Order of OR Expressions for THEN path Is Irrelevant" on page 167, we know that it does not help to put the most frequent true condition first (when ORing the conditions). If one of the conditions (say a=1) is much more likely and the code is in a loop, then switching to a SELECT statement costs 157 total path length (4399-4556) for the first use, but saves 872 execution path length (2579-1707) for each subsequent use.

TotlP	ExcuP	ScanP	Example

4399	2579	1820	(when a=1, b=0, c=0)
			
			if a=1 | b=1 | c=1
			   then do
			      end
			   else do
			      end

4450	1759	2691	(when a=1, b=0, c=0)
			
			if a=1
			   then do
			      end
			   else if b=1 | c=1
			      then do
			         end
			      else do
			         end

4556	1707	2849	(when a=1, b=0, c=0) √
			
			select
			   when a=1 then
			      do
			      end
			   when b=1 | c=1 then
			      do
			      end
			   otherwise
			   end

Order of AND Expressions for ELSE Path is Irrelevant: It is not faster to put the most frequent false condition first because all three conditions are evaluated anyway. However, this may influence the way you code (see "Use SELECT for AND-ELSE Path" on page 169).

TotlP	ExcuP	ScanP	Example

4486	2666	1820	`(when a=0, b=1, c=1)`
			
			`if a=1 & b=1 & c=1`
			`   then do`
			`      end`
			`   else do`
			`      end`

4486	2666	1820	`(when a=1, b=0, c=1)`
			
			`if a=1 & b=1 & c=1`
			`   then do`
			`      end`
			`   else do`
			`      end`

4483	2663	1820	`(when a=1, b=1, c=0)`
			
			`if a=1 & b=1 & c=1`
			`   then do`
			`      end`
			`   else do`
			`      end`

Use SELECT for AND-ELSE Path: From section "Order of AND Expressions for ELSE Path is Irrelevant" on page 168, we know that it does not help to put the most frequent false condition first (when ANDing the conditions). If one of the conditions (say a=0) is much more likely and the code is in a loop, then switching to a SELECT statement costs 47 total path length (4458-4505) for the first use, but saves 982 execution path length (2638-1656) for each subsequent use.

TotlP	ExcuP	ScanP	Example

4458	2638	1820	`if a=1 & b=1 & c=1`
			`   then do`
			`      end`
			`   else do`
			`      end`

4399	1708	2691	`if a=0`
			`   then do`
			`      end`
			`   else if b=1 & c=1`
			`      then do`
			`         end`
			`      else do`
			`         end`

4505	1656	2849	`select`
			`   when a=0 then`
			`      do`
			`      end`
			`   when b=1 & c=1 then`
			`      do`
			`      end`
			`   otherwise`
			`   end` √

Use Nested IFs In Place of SELECT: By using the nested IF, it saved 350 total path length (5296-4946) when the a=1 path was taken. Savings was similar when the b=1 path was taken and when the c=1 path was taken.

TotlP	ExcuP	ScanP	Example

4946
5473
6011
	1914
	2441
	2979
		3032
		3032
		3032
			(when a=1 path taken) √
			(when b=1 path taken)
			(when c=1 path taken)
			
			if a=1
			   then do
			      end
			   else if b=1
			      then do
			         end
			      else if c=1
			         then do
			            end

5296
5834
6383
	1944
	2482
	3031
		3352
		3352
		3352
			(when a=1 path taken)
			(when b=1 path taken)
			(when c=1 path taken)
			
			select
			   when a=1 then
			      do
			      end
			   when b=1 then
			      do
			      end
			   when c=1 then
			      do
			      end
			   end

Use IF Flag THEN ... ELSE: If you have a flag that is set to 1 for TRUE and 0 for FALSE, then using the second method saved 505 total path length. See “Use ′DO flag′ In Place of ′IF flag THEN′” when the IF statement only has a THEN path.

TotlP	ExcuP	ScanP	Example

2737	1321	1416	if a=1 /* given a=1 */
			   then do
			      end
			   else do
			      end

2232	906	1326	if a /* given a=1 */ √
			   then do
			      end
			   else do
			      end

Use ′DO flag′ In Place of ′IF flag THEN′: If you have a flag that is set to 1 for TRUE and 0 for FALSE, then the using the second method saved 388 total path length. See “Use IF Flag THEN ... ELSE” when the IF statement has both a THEN path and an ELSE path.

TotlP	ExcuP	ScanP	Example

1615	644	971	if a /* given a=1 */
			   then do
			      end

1227	697	530	do a /* given a=1 */ √ end

SELECT: Put Most Likely WHENs Higher: Each lower path costs 660 total path length (6634-5301)/2.

TotlP	ExcuP	ScanP	Example

5302
5968
6634
	1999
	2665
	3331
		3303
		3303
		3303
			`(when a=1 path is taken)` √
			`(when a=2 path is taken)`
			`(when a=3 path is taken)`
			
			`select`
			`  when a=1 then`
			`    do`
			`    end`
			`  when a=2 then`
			`    do`
			`    end`
			`  when a=3 then`
			`    do`
			`    end`
			`  end`

Working With Variables

To Initialize All Compound Variables, aaaa. = 5: Initializing just the stem saved 14342 total path length.

TotlP	ExcuP	ScanP	Example

15236	14175	1061	`do i = 1 to 10`
			`   aaaa.i = 5`
			`   end`

894	570	324	`aaaa. = 5` √

Use PARSE to Initialize Variables: Using PARSE to initialize the variables saved 871 total path length.

TotlP	ExcuP	ScanP	Example

2978	1835	1143	`aaa = 1`
			`bbb = 2`
			`ccc = 3`
			`ddd = 4`
			`eee = 5`

2107	1265	842	`parse value '1 2 3 4 5' with aaa bbb ccc ddd eee` √

Use Short Variable Names: Using the short variable name saved 48 total path length.

TotlP	ExcuP	ScanP	Example
657	417	240	`aaaaaa=1`
609	409	200	`a=1` √

Start Parts of Compound Name Tail With a Digit: In *vara.aaaa.bbbb*, *vara.* is the stem, and *aaaa* and *bbbb* might be variables, so REXX must do a variable look-up for *aaaa* and *bbbb*.

If *aaaa* and *bbbb* are not really variables, it was faster to use *vara.1aaa.1bbb* because the digit 1 caused REXX to suppresses the variable lookup. The example saved 19 total path length.

TotlP	ExcuP	ScanP	Example
871	549	322	`vara.aaaa.bbbb = data`
852	530	322	`vara.1aaa.1bbb = 'data'` √

Use PARSE VAR to Process Words in a Variable: Using PARSE VAR saved 3708 total path length.

TotlP	ExcuP	ScanP	Example

14700	13235	1465	`a = 'WORD1   WORD2 WORD3 WORD4 WORD5'`
			
			`do i = 1 to 'WORDS'(a)`
			`   x = 'WORD'(a,i)`
			`   end`

10992	9560	1432	`a = 'WORD1   WORD2 WORD3 WORD4 WORD5'` √
			
			`do i = 1 to 'WORDS'(a)`
			`   parse var a x a`
			`   end`

Other Tips

Set Default REXX Environment: address 'COMMAND': It may pay to set a new default REXX environment (as in the second example) rather than to keep overriding an existing REXX environment (as in the first example).

Setting a new default environment cost 482 total path length, and each time the new default was used saved 336 (4383-4047). The savings started with the second use.

TotlP	ExcuP	ScanP	Example
4383	4071	312	`address 'COMMAND' 'MAKEBUF'`
482 4047	217 4039	265 8	`address 'COMMAND'` `'MAKEBUF'` √

Built-in Functions Are Generally More Efficient.: Using the built in function saved 7631 total path length.

TotlP	ExcuP	ScanP	Example

9448	8299	1149	`n=10`
			
			`text = ''`
			`do n`
			`   text = text'*'`
			`   end`

1817	1232	585	`n=10` √ `text = 'COPIES'('*',n)`

Remove Loops When Possible: Removing the loop saved 8797 total path length.

TotlP	ExcuP	ScanP	Example

12413	10961	1452	`sumi = 0`
			`do i = 1 to 5`
			`   sumi = sumi + a.i`
			`   end`

3616	2821	795	`sumi = a.1 + a.2 + a.3 + a.4 + a.5` √

Combine Statements Where Logic Permits: This example sets b to a 4 digit number with leading zeros (with a=23, b will be set to 0023). Combining statements saved 665 total path length. See "Understand Available REXX Capabilities" for an even better way of performing the function.

TotlP	ExcuP	ScanP	Example
3532	2484	1048	`b='RIGHT'(a,4)` `b='TRANSLATE'(b,'0',' ')`
2867	2019	848	`b='TRANSLATE'('RIGHT'(a,4),'0',' ')` √

Understand Available REXX Capabilities: This is a refinement of "Combine Statements Where Logic Permits." The second example below takes advantage of a third parameter of the RIGHT function (the TRANSLATE was no longer needed). The savings was 868 total path length.

TotlP	ExcuP	ScanP	Example
2867	2019	848	`b='TRANSLATE'('RIGHT'(a,4),'0',' ')`
1999	1427	572	`b = 'RIGHT'(a,4,'0')` √

...=... Versus ...==...: The two comparison operators function as follows:

- '==' first check the length of the two operands. If the lengths are equal then a character compare is performed.
- '=' first tries a numeric compare. If both operands are numbers the comparison is done. If either is not a number then any leading blanks of both operands are removed and a character compare (right blank padding) is performed.

The net of the above is that they are different. In the example below the exact value of the operand was known so the strictly equal operand, '==', saved 4 total path length.

TotlP	ExcuP	ScanP	Example
1667	835	832	`if answer = 'YES' then nop    /* with answer = 'YES'  */`
1663	794	869	`if answer == 'YES' then nop   /* with answer = 'YES'  */` √

Use the Concatenation Operator Only When Necessary: When concatenation without an intervening blank is desired, removing the concatenation operator saved 93 total path length.

TotlP	ExcuP	ScanP	Example
1040	644	396	`a = 'pre'||line`
947	625	322	`a = 'pre'line` √

When concatenation with intervening blank is desired, removing the concatenation operator saved 410 total path length.

TotlP	ExcuP	ScanP	Example
1326	808	518	`a = 'pre'||' '||line`
916	601	315	`a = 'pre' line` √

Minimize Scanning of Unexecuted Code: Scanning of unexecuted code can cause significant unnecessary overhead. If your REXX program has code that is infrequently executed (but scanned), the infrequently executed code could be made into a routine and either externalized or moved to the bottom (this could require the practice of quoted function/subroutine calls; see "REXX Scanning and Executing" on page 157 and "Quoted Function, Subroutine Trade-off").

When the less likely event does occur (the code is executed) the path length would be longer, but with infrequent use this could be an excellent trade-off.

To get a feel for the cost of the scanning, look at the scan path length column in the measurement tables. You will notice that it is a fairly small percentage of the cost of commands that REXX passes to other environments (CMS for example). However, for REXX instructions, scan path length can be a significant percentage. The largest percentage improvement would be for programs that are heavy with REXX instructions and do not have many loops.

Quoted Function, Subroutine Trade-off: Use of the quoted function/subroutine suppresses a cache of the location of the function. Not using the quotes cost 49 execution path length (630-581) for the first invocation, but saved 34 execution path length (581-547) on subsequent invocations. Results will vary greatly with the functions used and sequence of execution. This comparison relates to execution path length so scan path length and total path length are not presented.

TotlP	ExcuP	ScanP	Example
	581		`x='QUEUED'()` • looks external to the program
	630		`x=queued()` • sets lookaside • causes scan to end of program (if not already scanned)
	547		`x=queued()` √ • uses lookaside

Enclose Nonvariable Data in Quotes: Adding the quotes saved 138 total path length (4879-4741). The quotes tell REXX that it is looking at a literal string, so there is no need for REXX to search to see if a variable (MAKEBUF for example) exists. The more variables used in the REXX program, the more significant the savings would be. The test exec had just 2 variables.

Here is another example:

```
address COMMAND LISTFILE fn ft fm          /* O.K.   */

address 'COMMAND' 'LISTFILE' fn ft fm      /* better */
```

You may not enclose REXX keywords in quotes. REXX keywords are in upper case in the REXX syntax diagrams:

```
'do' i = 1 'to' 3     /* does not work */

do i = 1 to 3         /* works         */
```

TotlP	ExcuP	ScanP	Example
4879	4427	452	address COMMAND MAKEBUF
4749	4316	433	address COMMAND 'MAKEBUF'
4741	4318	423	address 'COMMAND' 'MAKEBUF' √

Reduce Arithmetic Precision (Set by NUMERIC DIGITS): It took 1018 total path length to set NUMERIC DIGITS. For arithmetic in which numbers and results are low precision, dropping the precision to 5 made no difference. However, for calculations resulting in more significant digits (for example, calculating 1 / 3), decreasing the precision from 9 (the default) to 5 decreased total path length by 480 (2514-2034) for just this one calculation.

TotlP	ExcuP	ScanP	Example
1018	592	426	NUMERIC DIGITS n √
1250 1250 1250	959 959 959	291 291 291	x=1+1 /* given precision=5 */ x=1+1 /* given precision=7 */ x=1+1 /* given precision=9 */
2034 2274 2514	1739 1979 2219	295 295 295	x=1/3 /* given precision=5 */ √ x=1/3 /* given precision=7 */ x=1/3 /* given precision=9 */

Remove the Semicolon at the End of the Line: Removing the semicolon at end of the line saved 23 scan path length.

TotlP	ExcuP	ScanP	Example
671	447	224	x=1;
648	447	201	x=1 √

Exit a Loop Via LEAVE Instead of SIGNAL: Using LEAVE saved 1201 total path length.

TotlP	ExcuP	ScanP	Example	

11542	8651	2891	`do i = 1 to 5`
			`   if i=3 then signal done`
			`   end`
			`done:  say i`	

10341	8414	1927	`do i = 1 to 5`
			`   if i=3 then leave`
			`   end`
			`say i`
				√

Use ITERATE to Alter the Flow in a Loop: Using ITERATE saved 1500 total path length.

TotlP	ExcuP	ScanP	Example	

13392	11107	2285	`do i = 1 to 5`
			`   if i¬=3 then do`
			`      .`
			`      end`
			`   end`	

11892	10121	1771	`do i = 1 to 5`
			`   if i=3 then iterate`
			`   .`
			`   end`
				√

Invoke Code: Inline, Routine, Procedure, External Exec: From fastest to slowest:

TotlP	ExcuP	ScanP	Example	
1122	819	303	`y=y+1      /* inline code         */`	√

5267	2665	2602	`y=bump(y)  /* internal routine    */`
			• the routine was:
			`bump:return y+1`	

6171	3068	3103	`y=bump(y)  /* internal procedure  */`
			• the procedure was:
			`bump:procedure expose y`
			`return y+1`
			• One of the reasons that the internal procedure had a longer path length than the internal routine was that a new variable tree was set up for the internal procedure. This could provide the internal procedure with an advantage if the calling routine had lots of variables (because REXX would not have to search through all the calling routine variables when doing a variable lookup).	

73050	70431	2619	`y=bump(y)  /* external exec       */`
			• Where the external exec was:
			`/* */`
			`exit arg(1)+1`	

Use EXPOSE to Pass Parameter: When you are using a PROCEDURE, using EXPOSE to pass a parameter saved 823 total path length.

TotlP	ExcuP	ScanP	Example

5810	2393	3417	`stext = 'StringOfText'`
			`call sub stext`
			• the procedure was:
			`sub: procedure`
			`parse arg stext`
			`return`

4987	1866	3121	`stext = 'StringOfText'` √
			`call sub`
			• the procedure was:
			`sub: procedure expose stext`
			`return`

Abbreviate CP Commands and Parameters: Overall savings is 240 total path length (5028-4788).

TotlP	ExcuP	ScanP	Example
5028	4531	497	`address 'COMMAND' 'CP SPOOL READER HOLD'`
4956	4474	482	`address 'COMMAND' 'CP SP READER HOLD'`
4836	4379	457	`address 'COMMAND' 'CP SP R HOLD'`
4788	4341	447	`address 'COMMAND' 'CP SP R HO'` √

Try Alternate Approaches

- Try a different algorithm.
- Some things work faster in PIPES. PIPES tends to run faster when the I/O data is to be manipulated and pipeline filters can be used for the manipulation.
- Convert high use REXX programs to modules (ASSEMBLE or a high level language that is compiled).
- Use the REXX compiler if available.

Obtaining the Path Lengths

Using CP TRACE to Get CMS Path Length: CMS path lengths can be obtained by placing the following instructions with the code being measured:

```
address 'COMMAND' 'CP TRACE I NOTERM' /* start counting     */
address 'COMMAND' 'CP TRACE COUNT'    /* reset count        */
  (test A)
address 'COMMAND' 'CP TRACE COUNT'    /* display then reset */
  (test B)
address 'COMMAND' 'CP TRACE COUNT'    /* display then reset */
  (test C)
address 'COMMAND' 'CP TRACE COUNT'    /* display then reset */
address 'COMMAND' 'CP TRACE END ALL'  /* display and end    */
```

This tracing causes your virtual machine to run *much slower* because of the processing required to count the CMS instructions.

When the above example is run, the first two CP TRACE instructions do not generate any output. The first three measurements displayed “Trace count is nnnn” are for the three test cases respectively. The last displayed measurement can be ignored. The output is something like:

```
  (test case A running here)
Trace count is     7697
  (test case B running here)
Trace count is     7463
  (test case C running here)
Trace count is     7413
Trace count is     4860
Trace ended
```

The measurements above include path length associated with scanning and executing the instruction:

```
address 'COMMAND' 'CP TRACE COUNT'    /* display then reset */
```

Successive instructions of this type were tested from both EXECs and XEDIT macros to obtain the following trace overhead chart:

TotlP	ExcuP	ScanP	Example
4892	4289	603	(from an EXEC)
3331	2728	603	(from an XEDIT macro)

The path lengths are then obtained by subtracting the appropriate trace overhead from the measured results.

It is possible for the REXX application to fail during tracing. If instructions are being counted and the REXX program terminates without getting to

```
address 'COMMAND' 'CP TRACE END ALL'  /* display and end    */
```

your virtual machine will seem to be running *very slowly* because instructions are still being counted. To get your virtual machine back to normal, try

1. Pressing your virtual machine's break key (PA1 by default)
2. Enter

   ```
   TRACE END ALL
   ```

3. Enter

   ```
   BEGIN
   ```

Scanning Versus Execution Path Length: To obtain the breakout in virtual path length between scanning and execution[9], the following line of code was placed at the beginning of the file (after the initial REXX comment, /* */):

```
arg inparm; if inparm = 'EX' then x=queued()
```

- To obtain execution virtual path length, invoke the file with the single parameter, EX. This causes the above line to take the path that does

[9] Before reading this section you should understand the material presented in "REXX Scanning and Executing" on page 157

```
x=queued()
```

and the entire file is scanned[10]. Therefore path lengths for subsequent instructions in the application represent execution time (execution path length) because they were scanned when attempting to find the QUEUED function.

- To obtain the total virtual path length, invoke the file with no parameters. This does not force scanning as above, so path lengths of subsequent instructions contain both scanning and execution.

  Function and subroutine calls within the test case may cause scanning. This can be avoided by placing quotes around the name (as described in "REXX Scanning and Executing" on page 157).

- The scanning virtual path length is then calculated by taking the difference between total path length and execution path length.

[10] The file must not have its own routine called QUEUED because this would stop the scan.

Measurement Variability

Workload: FS7F0R

Hardware Configuration

Processor model:	9121-480
Processors used:	2
Storage	
Real:	224MB
Expanded:	32MB (all for MDC)
Tape:	3480 (Monitor)

DASD:

Type of DASD	*Control Unit*	*Number of Paths*	*PAGE*	*SPOOL*	*- Number of Volumes - TDSK*	*User*	*Server*	*System*
3390-2	3990-2	4	6	3	6			
3390-2	3990-3	4	10	3	2	16 R		2 R

Note: *R* or *W* next to the DASD counts means basic cache enabled or DASD fast write (and basic cache) enabled, respectively.

Communications:

Control Unit	*Number*	*Lines per Control Unit*	*Speed*
3088-08	1	NA	4.5MB

Software Configuration

Driver:	TPNS
Think time distribution:	Bactrian
CMS block size:	4KB

Virtual Machines:

Virtual Machine	*Number*	*Type*	*Machine Size/Mode*	*SHARE*	*RESERVED*	*Other Options*
SMART	1	RTM	16MB/370	3%	400	QUICKDSP ON
VTAMXA	1	VTAM/VSCS	64MB/XA	10000	512	QUICKDSP ON
WRITER	1	CP monitor	2MB/XA	10000		QUICKDSP ON
Users	1860	User	3MB/XC	100		

Measurement Discussion: Several changes were made to the FS7B CMS-intensive workload in order to improve measurement repeatability and the workload was renamed FS7F. See Appendix B, "Workloads" on page 191 for a description of the FS7F workload and how it differs from FS7B.

The measurement variability study done on VM/ESA Release 1.1 and documented in the *VM/ESA Release 2 Performance Report* was repeated on VM/ESA Release 2 using the FS7F workload in order to see to what extent measurement repeatability was improved and to provide an updated set of measurement variability data. The results, presented in this section, show that measurement repeatability has been improved by roughly a factor of two for most reported performance metrics.

The same FS7F measurement was repeated four times. Table 39 on page 182 shows the results for each individual measurement. Table 40 on page 184 shows the average, range, standard deviation, and percent standard deviation for these four runs. Range is the absolute value of the difference between the lowest and highest value observed. Standard deviation is calculated using the standard formula that assumes independent, normally distributed variables. Percent standard deviation is 100 times the standard deviation divided by the average.

If, for repeated measurements, one assumes that the values observed for any given metric are normally distributed, then that value should fall within one standard deviation of the average 68% of the time and within two standard deviations of the average 95% of the time.

Observe that some metrics show far more variability than others. For example, the percent standard deviation for external response time (AVG LAST (T)) is 2.6% while for internal throughput rate (ITR (H)) it is only 0.1%.

These results were obtained with VM/ESA Release 2. From informal observations, it is believed that VM/ESA Release 2.1 measurements show very similar run variabilities.

These results, of course, only apply to the measured environment: 9121-480, 224MB/32MB. However, for the FS7F0R workload, the relative variabilities of the various metrics are probably similar to these results in other environments as well. For example, AVG LAST (T) is almost certainly much more variable than ITR (H) regardless of what processor is used.

Table 39 (Page 1 of 2). Measurement variability of FS7F0R on the 9121-480—results				
Release	**VM/ESA 2**	**VM/ESA 2**	**VM/ESA 2**	**VM/ESA 2**
Run ID	**L25E1861**	**L25E1862**	**L25E1863**	**L25E1864**
Environment				
Real Storage	**224MB**	**224MB**	**224MB**	**224MB**
Exp. Storage	**32MB**	**32MB**	**32MB**	**32MB**
Users	**1800**	**1800**	**1800**	**1800**
VTAMs	**1**	**1**	**1**	**1**
VSCSs	**0**	**0**	**0**	**0**
Processors	**2**	**2**	**2**	**2**
Response Time				
TRIV INT	0.131	0.131	0.128	0.131
NONTRIV INT	0.473	0.485	0.481	0.489
TOT INT	0.361	0.368	0.364	0.370
TOT INT ADJ	0.315	0.323	0.322	0.326
AVG FIRST (T)	0.247	0.255	0.262	0.260
AVG LAST (T)	0.380	0.396	0.403	0.401
Throughput				
AVG THINK (T)	26.30	26.29	26.34	26.36
ETR	57.16	57.52	57.87	57.53
ETR (T)	65.42	65.47	65.48	65.35
ETR RATIO	0.874	0.879	0.884	0.880
ITR (H)	72.29	72.46	72.27	72.32
ITR	31.60	31.84	31.94	31.85
EMUL ITR	46.54	46.85	47.06	46.88
ITRR (H)	1.000	1.002	1.000	1.000
ITRR	1.000	1.008	1.011	1.008
Proc. Usage				
PBT/CMD (H)	27.664	27.603	27.673	27.655
PBT/CMD	27.667	27.646	27.641	27.699
CP/CMD (H)	9.591	9.557	9.595	9.571
CP/CMD	8.866	8.859	8.857	8.876
EMUL/CMD (H)	18.073	18.046	18.077	18.084
EMUL/CMD	18.801	18.787	18.784	18.823
Processor Util.				
TOTAL (H)	180.99	180.72	181.21	180.71
TOTAL	181.00	181.00	181.00	181.00
UTIL/PROC (H)	90.49	90.36	90.60	90.36
UTIL/PROC	90.50	90.50	90.50	90.50
TOTAL EMUL (H)	118.24	118.15	118.37	118.17
TOTAL EMUL	123.00	123.00	123.00	123.00
MASTER TOTAL (H)	90.26	90.15	90.42	90.13
MASTER TOTAL	90.00	90.00	90.00	90.00
MASTER EMUL (H)	53.48	53.43	53.57	53.47
MASTER EMUL	56.00	56.00	56.00	56.00
TVR(H)	1.53	1.53	1.53	1.53
TVR	1.47	1.47	1.47	1.47
Storage				
NUCLEUS SIZE (V)	2304KB	2304KB	2304KB	2304KB
TRACE TABLE (V)	400KB	400KB	400KB	400KB
WKSET (V)	83	83	84	84
PGBLPGS	47809	47770	47813	47805
PGBLPGS/USER	26.6	26.5	26.6	26.6
FREEPGS	4874	4862	4873	4882
FREE UTIL	0.95	0.95	0.95	0.94
SHRPGS	1173	1191	1163	1153

Table 39 (Page 2 of 2). Measurement variability of FS7F0R on the 9121-480—results				
Release	**VM/ESA 2**	**VM/ESA 2**	**VM/ESA 2**	**VM/ESA 2**
Run ID	**L25E1861**	**L25E1862**	**L25E1863**	**L25E1864**
Environment				
Real Storage	**224MB**	**224MB**	**224MB**	**224MB**
Exp. Storage	**32MB**	**32MB**	**32MB**	**32MB**
Users	**1800**	**1800**	**1800**	**1800**
VTAMs	**1**	**1**	**1**	**1**
VSCSs	**0**	**0**	**0**	**0**
Processors	**2**	**2**	**2**	**2**
Paging				
READS/SEC	632	629	632	624
WRITES/SEC	426	427	427	426
PAGE/CMD	16.172	16.129	16.172	16.068
PAGE IO RATE (V)	176.700	175.500	175.800	175.900
PAGE IO/CMD (V)	2.701	2.681	2.685	2.692
XSTOR IN/SEC	0	0	0	0
XSTOR OUT/SEC	0	0	0	0
XSTOR/CMD	0.000	0.000	0.000	0.000
FAST CLR/CMD	6.435	6.430	6.445	6.443
Queues				
DISPATCH LIST	40.78	37.39	40.23	39.89
ELIGIBLE LIST	0.00	0.00	0.02	0.00
I/O				
VIO RATE	624	622	624	622
VIO/CMD	9.538	9.500	9.529	9.519
RIO RATE (V)	398	395	396	395
RIO/CMD (V)	6.084	6.033	6.047	6.045
MDC READS	362	362	362	362
MDC WRITES	178	178	178	179
MDC MODS	145	145	145	145
MDC HIT RATIO	0.91	0.91	0.91	0.91
PRIVOPs				
PRIVOP/CMD	17.617	17.613	17.596	17.638
DIAG/CMD	26.921	26.853	26.851	26.903
DIAG 08/CMD	0.781	0.776	0.780	0.780
DIAG 14/CMD	0.025	0.024	0.025	0.025
DIAG 58/CMD	1.245	1.243	1.242	1.243
DIAG 98/CMD	1.179	1.148	1.159	1.154
DIAG A4/CMD	3.921	3.914	3.916	3.929
DIAG A8/CMD	1.873	1.883	1.884	1.883
DIAG 214/CMD	12.939	12.905	12.889	12.923
SIE/CMD	54.538	54.345	54.473	54.449
SIE INTCPT/CMD	37.631	37.498	37.587	37.570
FREE TOTL/CMD	63.740	63.845	63.789	63.907
VTAM Machines				
WKSET (V)	533	515	491	537
TOT CPU/CMD (V)	4.0252	3.9967	4.0384	4.0044
CP CPU/CMD (V)	1.7663	1.7395	1.7817	1.7599
VIRT CPU/CMD (V)	2.2588	2.2571	2.2568	2.2445
DIAG 98/CMD (V)	1.180	1.149	1.160	1.154
Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM				

Table 40 (Page 1 of 2). Measurement variability of FS7F0R on the 9121-480—statistics

Statistic	AVERAGE	RANGE	STD DEV.	% STD DEV.
Response Time				
TRIV INT	0.130	0.003	0.002	1.2
NONTRIV INT	0.482	0.016	0.007	1.4
TOT INT	0.366	0.009	0.004	1.1
TOT INT ADJ	0.322	0.011	0.005	1.4
AVG FIRST (T)	0.256	0.015	0.007	2.6
AVG LAST (T)	0.395	0.023	0.010	2.6
Throughput				
AVG THINK (T)	26.32	0.07	0.03	0.1
ETR	57.52	0.71	0.29	0.5
ETR (T)	65.43	0.13	0.06	0.1
ETR RATIO	0.879	0.01	0.00	0.5
ITR (H)	72.34	0.19	0.09	0.1
ITR	31.81	0.34	0.15	0.5
EMUL ITR	46.83	0.52	0.22	0.5
Proc. Usage				
PBT/CMD (H)	27.649	0.070	0.032	0.1
PBT/CMD	27.663	0.058	0.026	0.1
CP/CMD (H)	9.579	0.038	0.018	0.2
CP/CMD	8.865	0.019	0.009	0.1
EMUL/CMD (H)	18.070	0.038	0.017	0.1
EMUL/CMD	18.799	0.039	0.018	0.1
Processor Util.				
TOTAL (H)	180.91	0.50	0.24	0.1
TOTAL	181.00	0.00	0.00	0.0
UTIL/PROC (H)	90.45	0.24	0.12	0.1
UTIL/PROC	90.50	0.00	0.00	0.0
TOTAL EMUL (H)	118.23	0.22	0.10	0.1
TOTAL EMUL	123.00	0.00	0.00	0.0
MASTER TOTAL (H)	90.24	0.29	0.13	0.1
MASTER TOTAL	90.00	0.00	0.00	0.0
MASTER EMUL (H)	53.49	0.14	0.06	0.1
MASTER EMUL	56.00	0.00	0.00	0.0
TVR(H)	1.53	0.00	0.00	0.0
TVR	1.47	0.00	0.00	0.0
Storage				
WKSET (V)	84	1	1	0.7
PGBLPGS	47799	43	20	0.0
PGBLPGS/USER	26.6	0.10	0.05	0.2
FREEPGS	4873	20	8	0.2
FREE UTIL	0.95	0.01	0.01	0.5
SHRPGS	1170	38	16	1.4
Paging				
READS/SEC	629	8	4	0.6
WRITES/SEC	427	1	1	0.1
PAGE/CMD	16.135	0.104	0.049	0.3
PAGE IO RATE (V)	175.975	1.200	0.512	0.3
PAGE IO/CMD (V)	2.690	0.020	0.009	0.3
XSTOR IN/SEC	0	0	0	0.0
XSTOR OUT/SEC	0	0	0	0.0
XSTOR/CMD	0.000	0.000	0.000	0.0
FAST CLR/CMD	6.438	0.015	0.007	0.1
Queues				
DISPATCH LIST	39.57	3.39	1.50	3.8
ELIGIBLE LIST	0.01	0.02	0.01	200.0

Table 40 (Page 2 of 2). Measurement variability of FS7F0R on the 9121-480—statistics				
Statistic	**AVERAGE**	**RANGE**	**STD DEV.**	**% STD DEV.**
I/O				
VIO RATE	623	2	1	0.2
VIO/CMD	9.522	0.038	0.016	0.2
RIO RATE (V)	396	3	1	0.4
RIO/CMD (V)	6.052	0.051	0.022	0.4
MDC READS	362	0	0	0.0
MDC WRITES	178	1	1	0.3
MDC MODS	145	0	0	0.0
MDC HIT RATIO	0.91	0.00	0.000	0.0
PRIVOPs				
PRIVOP/CMD	17.616	0.042	0.017	0.1
DIAG/CMD	26.882	0.070	0.035	0.1
DIAG 08/CMD	0.779	0.005	0.002	0.3
DIAG 14/CMD	0.025	0.001	0.001	2.0
DIAG 58/CMD	1.243	0.003	0.001	0.1
DIAG 98/CMD	1.160	0.031	0.013	1.2
DIAG A4/CMD	3.920	0.015	0.007	0.2
DIAG A8/CMD	1.881	0.011	0.005	0.3
DIAG 214/CMD	12.914	0.050	0.022	0.2
SIE/CMD	54.451	0.193	0.080	0.1
SIE INTCPT/CMD	37.572	0.133	0.055	0.1
FREE TOTL/CMD	63.820	0.167	0.072	0.1
VTAM Machines				
WKSET (V)	519	46	21	4.0
TOT CPU/CMD (V)	4.0162	0.0417	0.0191	0.5
CP CPU/CMD (V)	1.7619	0.0422	0.0175	1.0
VIRT CPU/CMD (V)	2.2543	0.0143	0.0066	0.3
DIAG 98/CMD (V)	1.161	0.031	0.014	1.2
Note: T=TPNS, V=VMPRF, H=Hardware Monitor, Unmarked=RTM				

Appendixes

Appendix A. SFS Counter Data

The SFS counts and timings in this appendix are provided to supplement the information provided for the SFS measurements. These were acquired by issuing the QUERY FILEPOOL STATUS command once at the beginning of the measurement interval and once at the end.

The QUERY FILEPOOL STATUS information was obtained for each SFS file pool server and the CRR recovery server. The counts and timings for each server were added together.

The first section in the following table consists of the counters normalized by the number of commands (as determined by TPNS). The beginning values were subtracted from the ending values and divided by the number of commands in the interval. Counts and timings that have a value of zero for all measurements are not shown. A zero entry indicates that at least one occurrence was counted but the result of normalizing per command is so small that it rounds to zero. A description of the SFS counts and timings can be found in *SFS and CRR Planning, Administration, and Operation*.

The second section consists of derived relationships what were calculated from a combination of two or more individual counts or timings. See the glossary for definitions of these derived values.

Table 41 (Page 1 of 2). SFS counter data for measurements in this report.

RELEASE	VM/ESA 2	VM/ESA 2.1	VM/ESA 2	VM/ESA 2.1	VM/ESA 2.1
RUN ID	**S45S4562**	**S46S4561**	**L25S1620**	**L26S1620**	**L26S1563**
PROCESSOR	**9121-742**	**9121-742**	**9121-480**	**9121-480**	**9121-480**
REAL STORAGE	**1024MB**	**1024MB**	**224MB**	**224MB**	**224MB**
EXP. STORAGE	**1024MB**	**1024MB**	**32MB**	**32MB**	**32MB**
WORKLOAD	**FS7FMAXR**	**FS7FMAXR**	**FS7FMAXR**	**FS7FMAXR**	**FS7FMAXR**
USERS	**4560**	**4560**	**1620**	**1620**	**1560**
NORMALIZED BY COMMAND					
Close File Requests	0.3990	0.3961	0.3977	0.3943	0.3925
Commit Requests	0.0345	0.0344	0.0343	0.0344	0.0343
Connect Requests	0.0080	0.0081	0.0081	0.0080	0.0078
Delete File Requests	0.0969	0.0969	0.0964	0.0967	0.0966
Lock Requests	0.0245	0.0245	0.0246	0.0247	0.0246
Open File New Requests	0.0016	0.0016	0.0016	0.0016	0.0016
Open File Read Requests	0.2279	0.2281	0.2275	0.2270	0.2255
Open File Replace Requests	0.1480	0.1450	0.1474	0.1444	0.1441
Open File Write Requests	0.0214	0.0214	0.0213	0.0213	0.0212
Query File Pool Requests	0.0000	0.0000	0.0000	0.0000	0.0000
Query User Space Requests	0.0214	0.0214	0.0213	0.0213	0.0213
Read File Requests	0.1727	0.1729	0.1719	0.1720	0.1712
Refresh Directory Requests	0.0230	0.0229	0.0228	0.0229	0.0226
Rename Requests	0.0049	0.0049	0.0049	0.0049	0.0049
Unlock Requests	0.0246	0.0246	0.0246	0.0246	0.0246
Write File Requests	0.0708	0.0708	0.0703	0.0704	0.0703
Total File Pool Requests	1.2792	1.2736	1.2747	1.2686	1.2633
File Pool Request Service Time	31.0685	30.9320	32.4190	32.0883	28.6438
Local File Pool Requests	1.2792	1.2736	1.2747	1.2686	1.2633
Begin LUWs	0.5051	0.4995	0.5034	0.5005	0.4976
Agent Holding Time (msec)	56.9066	56.4179	103.0862	102.2541	90.9211
SAC Calls	6.1300	6.0484	6.0986	6.0821	6.0510
Catalog Lock Conflicts	0.0008	0.0007	0.0007	0.0007	0.0007
Total Lock Conflicts	0.0008	0.0007	0.0007	0.0007	0.0007
Lock Wait Time (msec)	0.0086	0.0076	0.0062	0.0071	0.0053
File Blocks Read	0.9647	0.9653	0.9609	0.9613	0.9563
File Blocks Written	0.6069	0.6076	0.6035	0.6043	0.6041
Catalog Blocks Read	0.4689	0.4608	0.5485	0.5450	0.4954
Catalog Blocks Written	0.2632	0.2528	0.2717	0.2768	0.2565
Control Minidisk Blocks Read	0.0000	0.0000	0.0000	0.0000	0.0000
Control Minidisk Blocks Written	0.0692	0.0674	0.0726	0.0726	0.0594
Log Blocks Written	0.5477	0.5409	0.5549	0.5579	0.5583
Total DASD Block Transfers	2.9206	2.8948	3.0122	3.0179	2.9301
BIO Requests to Read File Block	0.4314	0.4312	0.4321	0.4319	0.4281
BIO Requests to Write File Blocks	0.2273	0.2272	0.2276	0.2279	0.2272
BIO Requests to Read Catalog Blocks	0.4689	0.4608	0.5485	0.5450	0.4954
BIO Requests to Write Catalog Blocks	0.1759	0.1680	0.2132	0.2183	0.1995
BIO Requests to Read Cntrl Mdsk Blks	0.0000	0.0000	0.0000	0.0000	0.0000
BIO Requests to Write Cntrl Mdsk Blks	0.0025	0.0025	0.0025	0.0025	0.0025
BIO Requests to Write Log Blocks	0.4741	0.4682	0.4817	0.4843	0.4848
Total BIO Requests	1.7800	1.7580	1.9057	1.9099	1.8376
Total BIO Request Time (msec)	23.2969	22.4129	23.9293	23.8961	22.1915
I/O Requests to Read File Blocks	0.4205	0.3719	0.4262	0.3828	0.3804
I/O Requests to Write File Blocks	0.2469	0.2471	0.2477	0.2474	0.2479
I/O Requests to Read Catalog Blocks	0.4689	0.4608	0.5485	0.5450	0.4954
I/O Requests to Write Catalog Blocks	0.1878	0.1800	0.2218	0.2265	0.2075
I/O Requests to Read Cntrl Mdsk Blks	0.0000	0.0000	0.0000	0.0000	0.0000
I/O Requests to Write Cntrl Mdsk Blks	0.0047	0.0046	0.0049	0.0049	0.0048
I/O Requests to Write Log Blocks	0.4745	0.4686	0.4821	0.4847	0.4853
Total I/O Requests	1.8033	1.7330	1.9313	1.8912	1.8214

Table 41 (Page 2 of 2). SFS counter data for measurements in this report.					
RELEASE	**VM/ESA 2**	**VM/ESA 2.1**	**VM/ESA 2**	**VM/ESA 2.1**	**VM/ESA 2.1**
RUN ID	**S45S4562**	**S46S4561**	**L25S1620**	**L26S1620**	**L26S1563**
PROCESSOR	**9121-742**	**9121-742**	**9121-480**	**9121-480**	**9121-480**
REAL STORAGE	**1024MB**	**1024MB**	**224MB**	**224MB**	**224MB**
EXP. STORAGE	**1024MB**	**1024MB**	**32MB**	**32MB**	**32MB**
WORKLOAD	**FS7FMAXR**	**FS7FMAXR**	**FS7FMAXR**	**FS7FMAXR**	**FS7FMAXR**
USERS	**4560**	**4560**	**1620**	**1620**	**1560**
Get Logname Requests	0.0033	0.0033	0.0033	0.0033	0.0032
Get LUWID Requests	0.0033	0.0033	0.0033	0.0033	0.0032
Total CRR Requests	0.0066	0.0065	0.0065	0.0065	0.0064
CRR Request Service Time (msec)	0.0725	0.0769	0.0791	0.0779	0.0730
Log I/O Requests	0.0066	0.0065	0.0065	0.0065	0.0064
DERIVED RESULTS					
Agents Held	9.1	9.0	5.8	5.7	5.0
Agents In-call	5.0	5.0	1.8	1.8	1.6
Avg LUW Time (msec)	112.7	112.9	204.8	204.3	182.7
Avg File Pool Request Time (msec)	24.3	24.3	25.4	25.3	22.7
Avg Lock Wait Time (msec)	10.8	10.9	8.9	10.1	7.6
SAC Calls / FP Request	4.79	4.75	4.78	4.79	4.79
Deadlocks (delta)	0	0	0	0	0
Rollbacks Due to Deadlock (delta)	0	0	0	0	0
Rollback Requests (delta)	0	0	0	0	0
LUW Rollbacks (delta)	0	0	0	0	0
Checkpoints Taken (delta)	110	108	39	39	38
Checkpoint Duration (sec)	1.9	2.0	2.1	2.0	1.7
Seconds Between Checkpoints	17.5	17.8	49.5	49.5	50.8
Checkpoint Utilization	11.0	11.2	4.3	4.1	3.4
BIO Request Time (msec)	13.09	12.75	12.56	12.51	12.08
Blocking Factor (Blocks/BIO)	1.64	1.65	1.58	1.58	1.59
Chaining Factor (Blocks/IO)	1.62	1.67	1.56	1.60	1.61

Appendix B. Workloads

The workloads that were used to evaluate VM/ESA Release 2.1 are described in this appendix.

CMS-Intensive (FS7F)

Workload Description

FS7F[11] simulates a CMS user environment, with variations simulating a minidisk environment, an SFS environment, or some combination of the two. Table 42 shows the search-order characteristics of the two environments used for measurements discussed in this document.

Table 42. FS7F workload characteristics

Filemode	ACCESS	Number of Files	FS7F0R	FS7FMAXR
A	R/W	100	minidisk	SFS
B	R/W	0	minidisk	SFS
C	R/O	500	minidisk	SFS (DS)
D	R/W	500	minidisk	SFS
E	R/O	500	minidisk	SFS (DS)
F	R/O	500	minidisk	SFS (DS)
G	R/O	500	minidisk	SFS (DS)
S	R/O	*m*	minidisk	minidisk
Y	R/O	*n*	minidisk	minidisk

Note: *m* and *n* are the number of files normally found on the the S- and Y-disks respectively. (DS) signifies the use of VM Data Spaces.

The measurement environments have the following characteristics in common:

- The HELP disk has the FSTs saved in a shared segment.
- The CMSINST, VMLIB, and VMMTLIB shared segments are used.
- A Bactrian-distribution think time averaging 30 seconds is used. (See "Glossary of Performance Terms" on page 216 for an explanation of Bactrian distribution.)
- The workload is continuous in that scripts, repeated as often as required, are always running during the measurement period.
- Teleprocessing Network Simulator (TPNS) simulates users for the workload. TPNS runs in a separate processor and simulates LU2 terminals. User traffic travels between the processors through channel-attached 3745 communications controllers or 3088 multisystem channel communication units.

[11] FS7F workload is the same as FS7B (used in previous VM/ESA releases) except for enhancements to improve measurement repeatability. Some of the enhancements include placing screen clears and longer think times in fixed locations and improving the method of script selection so that the relative frequencies of script selection more closely match the requested frequencies.

FS7F Variations

Two FS7F workload variants were used for measurements, one for minidisk-based CMS users, and the other for SFS-based CMS users.

FS7F0R Workload: All filemodes are accessed as minidisk; SFS is not used. All of the files on the C-disk have their FSTs saved in a shared segment.

FS7FMAXR Workload: All file modes, except S and Y (which SFS does not support), the HELP minidisk, and T-disks that are created by the workload, are accessed as SFS directories. The CMSFILES shared segment is used. All read-only SFS directories are defined with PUBLIC READ authority and are mapped to VM data spaces. The read/write SFS directory accessed as file mode D is defined with PUBLIC READ and PUBLIC WRITE authority. The read/write SFS directories accessed as file modes A and B are private directories.

FS7F Licensed Programs

The following licensed programs were used in the FS7F measurements described in this document:

VS COBOL II Compiler and Library R3.2
Document Composition Facility V1.4
VS FORTRAN Compiler/Library/Debug V2R5
Assembler H V2.1 or IBM High Level Assembler
OS PL/I V2R3 Compiler & Library
C & PL/I Common Library V1R2
VTAM V3R4
NCP V5R4

Measurement Methodology

A calibration is made to determine how many simulated users are required to attain the desired processor utilization for the baseline measurement. That number of users is used for all subsequent measurements on the same processor and for the same environment.

The measurement proceeds as follows:

- All of the users are logged on by TPNS.
- A script is started for each user after a random delay of up to 15 minutes. (The random delay prevents all users from starting at once.)
- A stabilization period (the length depending on the processor used) is allowed to elapse so that start-up anomalies and user synchronization are eliminated.
- At the end of stabilization, measurement tools are started simultaneously to gather data for the measurement interval.
- At the end of the measurement interval, the performance data is reduced and analyzed.

FS7F Script Description

FS7F consists of 3 initialization scripts and 17 workload scripts. The LOGESA script is run at logon to set up the required search order and CMS configuration. Then users run the WAIT script, during which they are inactive and waiting to start the CMSSTRT script. The CMSSTRT script is run to stagger the start of user activity over a 15 minute interval. After the selected interval, each user starts running a general workload script. The scripts are summarized in Table 43.

Table 43. FS7F workload script summary

Script Name	**% Used**	**Script Description**
LOGESA	*	Logon and Initialization
WAIT	*	Wait state
CMSSTRT	*	Stagger start of user activity
ASM617A	5	Assemble (HASM) and Run
ASM627A	5	Assemble and Run
XED117	5	Edit a VS BASIC Program
XED127	10	Edit a VS BASIC Program
XED137	10	Edit a COBOL Program
XED147	10	Edit a COBOL Program
COB217	5	COBOL Compile
COB417	5	Run a COBOL Program
FOR217	5	VS FORTRAN Compile
FOR417	5	FORTRAN Run
PRD517	5	Productivity Aids Session
DCF517	5	Edit and Script a File
PLI317	5	PL/I Optimizer Session
PLI717	5	PL/I Optimizer Session
WND517A	8	Run Windows with IPL CMS
WND517AL	2	Run Windows with LOGON/LOGOFF
HLP517	5	Use HELP

Note: Scripts with an asterisk (*) in the "% Used" column are run only once each for each user during initialization.

The following are descriptions of each script used in the FS7F workload.

LOGESA: Initialization Script

LOGON userid
SET AUTOREAD ON
IF FS7F0R workload
THEN
  Erase extraneous files from A-disk
  Run PROFILE EXEC to access correct search order,
    SET ACNT OFF, SPOOL PRT CL D, and TERM LINEND OFF
ELSE
  Erase extraneous files from A-directory
  Run PROFILE EXEC to set correct search order, SET ACNT OFF,
    SPOOL PRT CL D, and TERM LINEND OFF
END
Clear the screen
SET REMOTE ON

WAIT: Ten-Second Pause
Leave the user inactive in a 10-second wait loop.

CMSSTRT: Random-Length Pause
Delay, for up to 15 minutes, the start for each user to prevent all users from starting scripts at the same time.

ASM617F: Assemble (HASM) and Run

QUERY reader and printer
SPOOL PRT CLASS D
XEDIT an assembler file and QQUIT
GLOBAL appropriate MACLIBs
LISTFILE the assembler file
Assemble the file using HASM (NOLIST option)
Erase the text deck
Repeat all the above except for XEDIT
Reset GLOBAL MACLIBs
Load the text file (NOMAP option)
Generate a module (ALL and NOMAP options)
Run the module
Load the text file (NOMAP option)
Run the module 2 more times
Erase extraneous files from A-disk

ASM627F: Assemble (F-Assembler) and Run

QUERY reader and printer
Clear the screen
SPOOL PRT CLASS D
GLOBAL appropriate MACLIBs
LISTFILE assembler file
XEDIT assembler file and QQUIT
Assemble the file (NOLIST option)
Erase the text deck
Reset GLOBAL MACLIBs
Load the TEXT file (NOMAP option)
Generate a module (ALL and NOMAP options)
Run the module
Load the text file (NOMAP option)
Run the module
Load the text file (NOMAP option)
Run the module
Erase extraneous files from A-disk
QUERY DISK, USERS, and TIME

XED117F: Edit a VS BASIC Program

XEDIT the program
Get into input mode
Enter 29 input lines
Quit without saving file (QQUIT)

XED127F: Edit a VS BASIC Program

Do a FILELIST
XEDIT the program
Issue a GET command
Issue a LOCATE command
Change 6 lines on the screen
Issue a TOP and BOTTOM command
Quit without saving file
Quit FILELIST
Repeat all of the above statements, changing 9 lines instead of 6 and without issuing the TOP and BOTTOM commands

XED137F: Edit a COBOL Program

Do a FILELIST
XEDIT the program
Issue a mixture of 26 XEDIT file manipulation commands
Quit without saving file
Quit FILELIST

XED147F: Edit a COBOL Program

Do a FILELIST
XEDIT the program
Issue a mixture of 3 XEDIT file manipulation commands
Enter 19 XEDIT input lines
Quit without saving file
Quit FILELIST

COB217F: Compile a COBOL Program

Set ready message short
Clear the screen
LINK and ACCESS a disk
QUERY link and disk
LISTFILE the COBOL program
Invoke the COBOL compiler
Erase the compiler output
RELEASE and DETACH the linked disk
Set ready message long
SET MSG OFF
QUERY SET
SET MSG ON
Set ready message short
LINK and ACCESS a disk
LISTFILE the COBOL program
Run the COBOL compiler
Erase the compiler output
RELEASE and DETACH the linked disk
QUERY TERM and RDYMSG
Set ready message long
SET MSG OFF
QUERY set
SET MSG ON
PURGE printer

COB417F: Run a COBOL Program

Define temporary disk space for 2 disks using an EXEC
Clear the screen
QUERY DASD and format both temporary disks
Establish 4 FILEDEFs for input and output files
QUERY FILEDEFs
GLOBAL TXTLIB
Load the program
Set PER Instruction
Start the program
Display registers
End PER
Issue the BEGIN command
QUERY search of minidisks
RELEASE the temporary disks
Define one temporary disk as another
DETACH the temporary disks
Reset the GLOBALs and clear the FILEDEFs

FOR217F: Compile 6 VS FORTRAN Programs

NUCXDROP NAMEFIND using an EXEC
Clear the screen
QUERY and PURGE the reader
Compile a FORTRAN program
Issue INDICATE commands
Compile another FORTRAN program
Issue INDICATE commands
Compile another FORTRAN program
Issue INDICATE command
Clear the screen
Compile a FORTRAN program
Issue INDICATE commands
Compile another FORTRAN program
Issue INDICATE commands
Compile another FORTRAN program
Clear the screen
Issue INDICATE command
Erase extraneous files from A-disk
PURGE the printer

FOR417F: Run 2 FORTRAN Programs

SPOOL PRT CLASS D
Clear the screen
GLOBAL appropriate text libraries
Issue 2 FILEDEFs for output
Load and start a program
Rename output file and PURGE printer
Repeat above 5 statements for two other programs, except
  erase the output file for one and do not issue spool printer
List and erase output files
Reset GLOBALs and clear FILEDEFs

PRD517F: Productivity Aids Session

Run an EXEC to set up names file for user
Clear the screen
Issue NAMES command and add operator
Locate a user in names file and quit
Issue the SENDFILE command
Send a file to yourself
Issue the SENDFILE command
Send a file to yourself
Issue the SENDFILE command
Send a file to yourself
Issue RDRLIST command, PEEK and DISCARD a file
Refresh RDRLIST screen, RECEIVE an EXEC on B-disk, and quit
TRANSFER all reader files to punch
PURGE reader and punch
Run a REXX EXEC that generates 175 random numbers
Run a REXX EXEC that reads multiple files of various sizes from
  both the A-disk and C-disk
Erase EXEC off B-disk
Erase extraneous files from A-disk

DCF517F: Edit and SCRIPT a File

XEDIT a SCRIPT file
Input 25 lines
File the results
Invoke SCRIPT processor to the terminal
Erase SCRIPT file from A-disk

PLI317F: Edit and Compile a PL/I Optimizer Program

Do a GLOBAL TXTLIB
Perform a FILELIST
XEDIT the PL/I program
Run 15 XEDIT subcommands
File the results on A-disk with a new name
Quit FILELIST
Enter 2 FILEDEFs for compile
Compile PL/I program using PLIOPT
Erase the PL/I program
Reset the GLOBALs and clear the FILEDEFs
QUERY virtual devices
TELL yourself one pass of script run

PLI717F: Edit, Compile, and Run a PL/I Optimizer Program

Copy and rename the PL/I program and data file from C-disk
XEDIT data file and QQUIT
XEDIT a PL/I file
Issue RIGHT 20, LEFT 20, and SET VERIFY ON
Change two lines
Change filename and file the result
Compile PL/I program using PLIOPT
Set two FILEDEFs and QUERY the settings
Issue GLOBAL for PL/I transient library
Load the PL/I program (NOMAP option)
Start the program
Type 8 lines of one data file
Erase extraneous files from A-disk
Erase extra files on B-disk
Reset the GLOBALs and clear the FILEDEFs
TELL yourself one pass of script run

WND517F: Use Windows

SET FULLSCREEN ON
TELL yourself a message to create window
QUERY DASD and reader
Forward 1 screen
TELL yourself a message to create window
Drop window message
Scroll to top and clear window
Backward 1 screen
Issue a HELP WINDOW and choose Change Window Size
QUERY WINDOW
Quit HELP WINDOWS
Change size of window message
Forward 1 screen
Display window message
TELL yourself a message to create window
Issue forward and backward border commands in window message
Position window message to another location
Drop window message
Scroll to top and clear window
Display window message
Erase MESSAGE LOGFILE
IPL CMS
SET AUTOREAD ON
SET REMOTE ON

WND517FL: Use Windows with LOGON, LOGOFF

SET FULLSCREEN ON
TELL yourself a message to create window
QUERY DASD and reader
Forward 1 screen
TELL yourself a message to create window
Drop window message
Scroll to top and clear window
Backward 1 screen
Issue a help window and choose Change Window Size
QUERY WINDOW
Quit help windows
Change size of window message
Forward 1 screen
Display window message
TELL yourself a message to create window
Issue forward and backward border commands in window message
Position window message to another location
Drop window message
Scroll to top and clear window
Display window message
Erase MESSAGE LOGFILE
LOGOFF user and wait 60 seconds
LOGON user on original GRAF-ID
SET AUTOREAD ON
SET REMOTE ON

HLP517F: Use HELP and Miscellaneous Commands

Issue HELP command
Choose HELP CMS
Issue HELP HELP
Get full description and forward 1 screen
Quit HELP HELP
Choose CMSQUERY menu
Choose QUERY menu
Choose AUTOSAVE command
Go forward and backward 1 screen
Quit all the layers of HELP
RELEASE Z-disk
Compare file on A-disk to C-disk 4 times
Send a file to yourself
Change reader copies to two
Issue RDRLIST command
RECEIVE file on B-disk and quit RDRLIST
Erase extra files on B-disk
Erase extraneous files from A-disk

IBM Office Benchmark (IOB V2.1)

Workload Description

The IBM Office Benchmark (IOB) Version 2.1 is a corporate-wide benchmark designed to measure generic office system performance. It consists of the office user definition, the databases for calendars, the documents, the mail, and the work the office users do.

The IOB measurements included in this report use the DisplayWrite/370 2.1.0 and the OfficeVision/VM 1.2.0 GA level licensed programs.

Measurement Methodology

A calibration is made to determine how many simulated users are required to attain the desired processor utilization for the baseline measurement. That number of users is used for all subsequent measurements on the same processor and for the same environment.

The measurement proceeds as follows:

- All of the users are logged on by TPNS and reach the OfficeVision main menu (the A00 screen).
- After a random delay of up to 10 minutes, each user selects a script and starts. (The random delay prevents all users from starting at once).
- A stabilization period (typically 35 minutes) is allowed to elapse so that start-up anomalies and user synchronization are eliminated.
- At the end of stabilization, measurement tools are started simultaneously to gather data for the measurement interval.
- At the end of the measurement interval, the performance data is reduced and analyzed.

After the measurement data are analyzed and qualified for IOB certification, the data are submitted for certification. All of the measurements in this report were certified as valid IOB measurements.

The IOB workload does not aim for a specific think time or use a certain think time distribution. Instead, the think time is dictated by the IOB workload. The think time includes an average two second delay between commands issued by TPNS, the built-in think times that are part of the IOB scripts, and the IOB script scheduling algorithm. When users finish running a script, the script scheduling algorithm calculates how much time was spent running the script, subtracts this number from ten minutes, and delays the user for the resulting amount of time. Thus, if a script completed in 7.9 minutes, the user would be delayed for 2.1 minutes before starting the next script and this time would be included in the user′s think time.

IOB Script Descriptions

The IOB workload consists of nine scripts (scenarios). These scripts are listed in Table 44 with their defined use factor.

Table 44. IOB workload script summary

Script Name	% Used	Script Description
VMB2LML	17	Send Note and Process Light Mail
VMB2HML	17	Send Note and Process Heavy Mail
VMB2VCAL	13	View Individual Calendar
VMB2UCAL	13	Update Individual Calendar
VMB2DIR	20	View User Directory
VMB2CDOC	7	Create Small Text Document
VMB2UDOC	7	Revise Small Text Document
VMB2EB	3	Enter/Exit Office
VMB2ONOF	3	Logoff/Logon System

The following is the list of tasks in each script within the IOB workload.

Send Note and Process Light Mail

- Create a note and send the note to two users.
- View the note log.
- View the first item, a note.
- Delete the first item, a note.
- Open Mail and View the In-Basket.
- View the first item, a note.
- Delete the first item, a note.

Send Note and Process Heavy Mail

- Create a note and send the note to two users.
- View the note log.
- View the first item, a note.
- Delete the first item, a note.
- Open Mail and View the In-Basket.
- View the first item, a note.
- Forward the first item to another user with an attachment.
- Delete the original first item, a note.
- View the eighth item in the mail list, a two page document.
- Print the document.

View Individual Calendar

- View the user's calendar for Wednesday of a defined week.

Update Individual Calendar

- View the user's calendar for Wednesday of a defined week.
- Delete a meeting.
- Add a meeting.

View User Directory

- Search the user directory based on a random user name and view the person's telephone number.

Create Small Text Document

- Get a pre-stored document format.
- Key in a two-page document.
- Save the document.
- Print the document.
- Delete the document.

Revise Small Text Document

- Open a two-page document for revision.
- Move one paragraph.
- Delete one paragraph.
- Insert one paragraph.
- Save the altered document.
- Send the document to three users.

Enter/Exit Office

- Enter the office software environment.
- Exit the office software environment.

Logon/Logoff System

- Log off from the system.
- Log back onto the system and enter the office environment.

VSE Guest (PACE)

PACE is a synthetic VSE batch workload consisting of 7 unique jobs representing the commercial environment. This set of jobs is replicated *c* times, producing the PACEX*c* workload, in order to scale the load to the environment being tested. The most commonly-used multiples are PACEX4, PACEX6, and PACEX8.

The seven jobs are as follows:

- Y*n*DL/1
- Y*n*SORT
- Y*n*COBOL
- Y*n*BILL
- Y*n*STOCK
- Y*n*PAY
- Y*n*FORT

There are *c* copies of these jobs used in the PACEX*c* workload. They are differentiated by the *n* digit in the name (*n* having a value from 1 to *c*).

The programs, data, and work space for the jobs are all maintained by VSAM on separate volumes.

The VSE system is configured with the full complement of 12 static partitions (BG, and F1 through FB). F4 through FB are the partitions used to run the workload batch jobs for PACEX8; a subset of these partitions are used for smaller PACE variations.

The partitions are configured identically except for the job classes. The jobs and the partition job classes are configured so that the jobs are equally distributed over the partitions and so that, at any one time, the jobs currently running are a mixed representation of the 7 jobs.

When a workload is ready to run, the following preparatory steps are taken:

- CICS/ICCF is shut down
- VTAM is shut down
- The LST queue is emptied (PDELETE LST,ALL)

Once performance data gathering is initiated for the system (hardware instrumentation, CP MONITOR, RTM), the workload is started by releasing all of the batch jobs into the partitions simultaneously using the POWER command, PRELEASE RDR,*Y. The start time is noted.

As the workload nears completion, various partitions will finish the work allotted to them. The finish time for both the first and last partitions is noted. The difference between these two times should not be more than about 10% of the total elapsed time. If it is more, the jobs and partitions have to be adjusted to get a more even work distribution.

At workload completion, the ITR can be calculated by dividing the number of batch jobs by processor busy time. The processor busy time is calculated as elapsed (wall clock) time multiplied by processor busy percent divided by 100.

VSE/ESA 1.2.0 was used in the measurements to gather data for the VSE guest measurements in this report.

Appendix C. Configuration Details

Named Saved Segments / Systems

CMS allows the use of saved segments for shared code. Using saved segments can greatly improve performance by reducing end users' working set sizes and thereby decreasing paging. The environments in this report used the following saved segments:

CMS	Contains the CMS nucleus and file status tables (FSTs) for the S- and Y-disks.
CMSFILES	Contains the SFS server code modules DMSDAC and DMSSAC
CMSINST	Contains the execs-in-storage segment
VMLIB	Contains the CSL code
VMMTLIB	Contains the CMS multitasking code
HELP	Contains FSTs for the HELP disk
GOODSEG	Contains FSTs for the C disk
FORTRAN	This segment space has 2 members: DSSVFORT for the FORTRAN compiler and FTNLIB20 for the library composite modules.
DSMSEG4B	Contains DCF (Document Composition Facility)
OFSSEG	Contains OV/VM user functions
EPUYSSEG	Contains OV/VM mailbox manager code
DW370210	Contains the DW370 module
DDDCL210	Contains the DW370 compiled CLISTS
DW362	Contains FSTs for the DW/370 362 disk
ADM399	Contains FSTs for the OV/VM 399 disk
GCSXA	Contains the GCS nucleus
VTAMXA	Contains the VTAM code

Server Options

SFS DMSPARMS This section lists the start-up parameter settings used by each of the SFS servers. The start-up parameters determine the operational characteristics of the file pool server. The SFS servers used the following DMSPARMS file:

```
ADMIN OPERATOR MAINT U3 MARK
FILEPOOLID fp_name
NOBACKUP
NOFORMAT
USERS  nnnn
FULLDUMP
SAVESEGID CMSFILES
MSGS
ACCOUNT
```

For all SFS measurements, the SAVESEGID is specified to identify the segment containing the file pool server runnable code. The USERS parameter is used by the SFS server to configure itself with the appropriate number of user agents and buffers. It is recommended that USERS be set to the administrator's best estimate of the maximum number of logged-on virtual machines that will be using the file pool during peak usage. The ratio of logged-on users to active users varies greatly on actual production machines.

For more information on SFS and SFS tuning parameters, see the *SFS and CRR Planning, Administration, and Operation* manual or the *VM/ESA Release 2.1 Performance* manual.

CRR DMSPARMS This section lists the start-up parameter settings used by the CRR recovery server. The start-up parameters determine the operational characteristics of the CRR recovery server. The CRR server uses the following DMSPARMS file:

```
ADMIN OPERATOR MAINT U3 MARK
NOBACKUP
FULLDUMP
SAVESEGID CMSFILES
FILEPOOLID fp_name
ACCOUNT
NOFORMAT
CRR
LUNAME lu_name
MSGS
```

For more information on CRR and CRR tuning parameters, see the *SFS and CRR Planning, Administration, and Operation* manual or the *VM/ESA Release 2.1 Performance* manual.

Appendix D. Creating a Logical Segment for CMS Modules

This topic covers the effects of increased paging and ways to reduce it. This is an enhanced version of a section from Chapter 3 of the *VM/ESA Performance* manual.

Preventing Oversized Working Sets

As more function has been added to CMS in new releases, the size of the CMS nucleus has grown with the added code. To prevent the CMS saved system from requiring 3MB of storage below the virtual 16M line, the size of the CMS nucleus was reduced by removing 16 infrequently called modules from the nucleus and putting them on the CMS S-disk. The removed modules handle various CMS QUERY and CMS SET commands. They are listed in Figure 18 on page 213.

Now when a CMS user first runs a QUERY or SET command handled by one of the new disk-resident modules, the appropriate module is made a nucleus extension to that user′s virtual machine through the NUCXLOAD function. NUCXLOAD loads the module into free storage in the user′s virtual machine where it remains for the duration of the user′s CMS session or until explicitly dropped. Subsequent calls to the module will not require reading the module from DASD because it is already in free storage. This lessens the performance penalty to the CMS user who calls a QUERY or SET command now handled by one of the 16 modules removed from the CMS nucleus.

A side effect of removing modules from the CMS nucleus (and the corresponding saved system copy of the CMS nucleus) is that pages that were formerly shared system-wide by CMS users are now private in those virtual machines that use them. Virtual machines can experience increased working set sizes which will lead to increased loads on the VM system′s paging subsystem. This increased paging load can be substantial if widely used EXECs and programs issue QUERY or SET commands that are handled by the modules removed from the CMS nucleus.

Four of these 16 modules are called by certain CMS functions and are therefore more likely to be used:

- DMSSEC handles the SET COMDIR NAMES command. SET COMDIR NAMES is issued in the SYSPROF EXEC distributed with CMS in VM/ESA.
- DMSSEF handles CMS fullscreen and windowing SET commands. Invoking the CMS HELP command will cause this module to be used. It also handles the the SET REMOTE ON command. SET REMOTE ON is used by many VM/VTAM customers to reduce the amount of data transmitted in full screen operations (like XEDIT and FULLSCREEN CMS). DMSSEF adds about 3 pages to the working set size.
- DMSQRF handles CMS fullscreen and windowing QUERY commands. OfficeVision/VM issues a QUERY FULLSCREEN and, if necessary, a SET FULLSCREEN SUSPEND (handled by DMSSEF) in many of its routines. The CMS HELP command also will cause this module to be used. DMSQRF adds about 3 pages to the working set size.

- DMSQRT handles QUERY CMSTYPE and QUERY CMSLEVEL. The FILELIST EXEC issues a QUERY CMSTYPE command. Many applications use QUERY CMSLEVEL to determine feature availability.

Your installation may also be making widespread use of some of the remaining 12 modules. You can check this by obtaining a NUCXMAP from a sampling of your CMS users to see if some of these modules show up frequently in the NUCXMAP output.

Using Logical Segment Support

VM/ESA logical segment support can provide relief from the increased working set size. By placing the removed modules into a logical segment, all users on the system can again use shared storage instead of private storage for those modules.

When modules are placed into a CMS logical segment and made available to the virtual machine by way of the SEGMENT LOAD command, the modules are automatically made nucleus extensions to the CMS nucleus in the virtual machine. The pages occupied by the modules are shared among all users of the logical segment instead of private pages as is the case when the logical segment is not used.

A potential location for the logical segment containing these modules is in the HELPINST physical segment which contains the logical segments named HELP and CMSINST. The following procedure assumes that HELPINST was installed as documented in the *VM/ESA Installation Guide*.

Note: Most of the 16 modules removed from the CMS nucleus must reside below the virtual 16MB line. The HELPINST physical segment was chosen for adding the logical segment to contain these modules because it is documented as being installed and saved below the 16MB virtual line in the *VM/ESA Installation Guide*. The segment containing the modules could be in any physical segment saved below the 16MB line.

CMS logical segment support enhances CP′s capabilities with respect to segments in several ways. One enhancement is that a logical saved segment can contain different types of program objects, such as modules, text files, execs, callable services libraries, language information, and user-defined objects, or minidisk information. Another enhancement is that you can save several different logical segments into a physical saved segment and allow end-users to access only the specific logical saved segments that they need rather than the entire physical saved segment. (See the section on saved segment planning considerations in the *VM/ESA Planning and Administration* book for more information.)

Creating a Saved Segment Containing HELP, INSTSEG and CMS Modules

The following example demonstrates how to add a new logical segment to contain the modules removed from the CMS nucleus into the previously defined HELPINST segment. As previously noted, most of the modules that were removed from the CMS nucleus must be saved below the 16MB line. Therefore, the physical segment containing the new logical segment must reside below the 16MB line.

The steps for adding a new logical segment to the HELPINST physical segment (PSEG) are as follows:

1. Logon to the MAINT userid
2. Enter the command

   ```
   VMFSGMAP SEGBLD ESASEGS SEGBLIST
   ```

   to view the segment map.
3. Locate the line for the HELPINST segment that was predefined by IBM. Move the cursor onto that line and press PF4=Chg_Obj.
4. The display should be modified to add the "PROD(LSEG CMSQRY)" into the BLDPARMS section. Update the OBJDESC description comment also. The display should look like this:

```
                              Change Segment Definition
                                                             Lines 1 to 13 of 13

 OBJNAME....:  HELPINST
 DEFPARMS...:  COO-CFF SR
 SPACE......:
 TYPE.......:  PSEG
 OBJDESC....:  CMSINST AND HELP AND CMSQRY LSEGS
 OBJINFO....:
 GT_16MB....:  NO
 DISKS......:
 SEGREQ.....:
 PRODID.....:  6VMVMA21 CMS
 BLDPARMS...:  PPF(ESA CMS DMSSBINS) PPF(ESA CMS DMSSBHLP) PROD(LSEG
            :  CMSQRY)



 F1=Help     F2=Get Obj   F3=Exit      F4=Add Line  F5=Map        F6=Chk MEM
 F7=Bkwd     F8=Fwd       F9=Retrieve F10=Seginfo   F11=Adj MEM   F12=Cancel
====>
```

Figure 17. Modifying HELPINST to contain CMSQRY LSEG

5. After adding to the BLDPARMS, press F5 to return to the MAP.
6. From the Segment Map screen, press F5=FILE to file the changes and exit VMFSGMAP.
7. Create the CMSQRY LSEG file

   The CMSQRY LSEG file contains the information for the new logical segment that will contain the modules that were removed from the CMS nucleus and made disk-resident. Create the CMSQRY LSEG file on MAINT′s 191 A-disk. It needs to be on the A-disk because when VMFBLD creates the PSEG file, it includes PROFILE/EPIFILE options for CMSINST and HELP that cause disks other than A, S, Y and the VMSES/E disks to be released. This is all right in this case because we desire to have the modules from the 190 S-disk loaded. But the releasing of the disks means that the CMS local modification disk (MAINT′s 3C4, by default) will not be accessed. Use the following modules from the S-disk when building CMSQRY segment:

```
* LOAD THESE MODULES FROM S DISK INTO AN LSEG
MODULE DMSQRF S ( SYSTEM )
MODULE DMSQRT S ( SYSTEM )
MODULE DMSSEF S ( SYSTEM )
MODULE DMSQRV S ( SYSTEM )
MODULE DMSQRC S ( SYSTEM )
MODULE DMSQRE S ( SYSTEM )
MODULE DMSQRG S ( SYSTEM )
MODULE DMSQRH S ( SYSTEM )
MODULE DMSQRN S ( SYSTEM )
MODULE DMSQRP S ( SYSTEM )
MODULE DMSQRQ S ( SYSTEM )
MODULE DMSQRU S ( SYSTEM )
MODULE DMSQRW S ( SYSTEM )
MODULE DMSQRX S ( SYSTEM )
MODULE DMSSEC S ( SYSTEM )
MODULE DMSSML S ( SYSTEM )
```

Figure 18. CMSQRY LSEG file

The first four modules in the list (DMSQRF, DMSSEF, DMSQRT, and DMSQRV) have been placed adjacent to each other in the list in hopes that there will be locality of reference for the pages holding those modules in an OV/VM environment. If you are familiar with other applications on your system that do frequent SET or QUERY commands handled by other modules in the list, you might also try putting those modules adjacent to each other in the list. This is a fine-tuning point that is optional. The largest benefit is obtained from placing the modules in this segment in order that the pages may be shared.

8. Create a local modification to the SYSPROF EXEC using VMSES/E procedures to cause the CMSQRY segment to be used. Refer to the *VM/ESA Service Guide* in the chapter entitled "Installing Local Service and Modifications" for information on making a local modification to the source file SYSPROF $EXEC.

   Where the SYSPROF EXEC has:

   ```
   'RTNLOAD * (FROM VMLIB SYSTEM GROUP VMLIB'
   ```

   Insert the following line immediately before the RTNLOAD statement:

   ```
   'SEGMENT LOAD CMSQRY (SYSTEM' /* Get shared QUERY and SET modules */
   ```

   The result should look like this:

   ```
   ⋮
   'SEGMENT LOAD CMSQRY (SYSTEM' /* Get shared QUERY and SET modules */
   'RTNLOAD * (FROM VMLIB SYSTEM GROUP VMLIB'
   ⋮
   ```

   It is important that the SEGMENT LOAD CMSQRY statement appear before any other CMS SET or CMS QUERY commands that you might add via local modification of the SYSPROF EXEC in order to get the benefits the segment provides.

Follow the instructions in the *VM/ESA Service Guide* to apply your local modification and to get the updated SYSPROF EXEC on the 190 and 490 minidisks.

9. IPL the 190 minidisk and prohibit the SYSPROF EXEC from running and also the CMSINST segment from loading:

   ```
   IPL 190 CLEAR PARM NOSPROF INSTSEG NO
   ```

10. Prohibit execution of the PROFILE EXEC:

    ```
    ACCESS (NOPROF
    ```

11. Reserve the storage where you will be loading the data to be saved:

    ```
    SEGMENT RESERVE HELPINST
    ```

12. Run the PROFILE EXEC:

    ```
    EXEC PROFILE
    ```

13. Execute the VMFBLD EXEC to cause the HELPINST segment to be rebuilt.

    ```
    VMFBLD PPF SEGBLD ESASEGS SEGBLIST HELPINST (ALL
    ```

14. Use VMFVIEW to view the build message log.

    ```
    VMFVIEW BUILD
    ```

    You should see the message:

    ```
    VMFBDS2003W The SYSTEM SEGID D(51D) file has been changed and must
    be moved to the S disk.
    ```

    in the build message log. Copy the SYSTEM SEGID file from the 51D minidisk to both the 190 and 490 and ensure that it has a filemode of **2**.

15. Resave the CMS NSS in order to regain the S-stat.

    Because the SYSTEM SEGID file was altered on the 190 disk, it is necessary to resave the CMS named saved system to regain the shared S-stat. Use the procedures appropriate for your environment (for example: SAMPNSS CMS followed by IPL 190 CLEAR PARM SAVESYS CMS).

    If some CMS users on the system are unable to load this segment because of conflicts with their Page Allocation Table (PAT) or a conflict with another saved segment that they need, then those users will still have private copies of the any individual modules that get dynamically loaded via NUCXLOAD because of a QUERY or SET command that they enter.

The only change that the user might see as a result of this process is after implementing this new logical segment, the response to a NUCXMAP command is likely to be longer than end users typically see. A typical response to NUCXMAP just after IPLing CMS and after the SYSPROF runs, will be similar to the following. While longer, the response is perfectly valid:

```
Ready;
nucxmap (seginfo
Name      Entry    Userword Origin   Bytes    Amode Segname
DMSSEF    00CCDF80 00000000 00CCDF80 00002AA0   ANY CMSQRY
NAMEFSYS  00E7FB28 00DE22C8 00E7FB28 00000000    31
NAMEFIND  00E7FB28 00DE2930 00E7FB28 00000000    31
DMSSEC    00CCD678 00000000 00CCD678 00000908   ANY CMSQRY
NAMEFUSE  00E7FB28 00DA5B58 00E7FB28 00000000    31
DMSQRC    00CBE1B0 00000000 00CBE1B0 00000440    31 CMSQRY
DMSQRE    00CBE5F0 00000000 00CBE5F0 00001188    31 CMSQRY
DMSQRF    00CBF778 00000000 00CBF778 00002E90   ANY CMSQRY
DMSQRG    00CC2608 00000000 00CC2608 00002510   ANY CMSQRY
DMSQRH    00CC4B18 00000000 00CC4B18 00001338   ANY CMSQRY
DMSQRN    00CC5E50 00000000 00CC5E50 00000408    31 CMSQRY
DMSQRP    00CC6258 00000000 00CC6258 00002DB0    31 CMSQRY
DMSQRQ    00CC9008 00000000 00CC9008 000017A8    31 CMSQRY
DMSQRT    00CCA7B0 00000000 00CCA7B0 00000EE0    31 CMSQRY
DMSQRU    00CCB690 00000000 00CCB690 00000890    24 CMSQRY
DMSQRV    00CCBF20 00000000 00CCBF20 00000850    31 CMSQRY
DMSQRW    00CCC770 00000000 00CCC770 00000520    31 CMSQRY
DMSQRX    00CCCC90 00000000 00CCCC90 000009E8    24 CMSQRY
DMSSML    00CD0A20 00000000 00CD0A20 00000B20    31 CMSQRY
PIPMOD    040C1EA0 00000000 040C1EA0 0003A6A0    31 PIPES
Ready;
```

Glossary of Performance Terms

Many of the performance terms use postscripts to reflect the sources of the data described in this document. In all cases, the terms presented here are taken directly as written in the text to allow them to be found quickly. Often there will be multiple definitions of the same data field, differing only in the postscript. This allows the precise definition of each data field in terms of its origins. The postscripts are:

<none>. No postscript indicates that the data are obtained from the VM/ESA Realtime Monitor.

(H). Denotes data obtained from the internal processor instrumentation tools.

(Q). Denotes data from the SFS QUERY FILEPOOL STATUS command.

Server. Indicates that the data are for specific virtual machines, (for example SFS, CRR, or VTAM/VSCS). If there is more than one virtual machine of the same type, these data fields are for all the virtual machines of that type.

(T). Identifies data from the licensed program, Teleprocessing Network Simulator (TPNS).

(V). Denotes data from the licensed program VM Performance Reporting Facility.

The formulas used to derive the various statistics are also shown here. If a term in a formula is in italics, such as *Total_Transmits*, then a description of how its value is derived is provided underneath the formula. If a term is not in italics, such as SFSTIME, then it has an entry in the glossary describing its derivation.

Absolute Share. An ABSOLUTE share allocates to a virtual machine an absolute percentage of all the available system resources.

Agent. The unit of sub-dispatching within a CRR or SFS file pool server.

Agents Held. The average number of agents that are in a Logical Unit of Work (LUW). This is calculated by:

$$\frac{1}{1000} \times \sum_{f \in \text{filepools}} \frac{\mathit{Agent\_Holding\_Time}_f}{\text{SFSTIME}_f}$$

Agent_Holding_Time is from the QUERY FILEPOOL STATUS command.

Agents In Call. The average number of agents that are currently processing SFS server requests. This is calculated by:

$$\frac{1}{1000} \times \sum_{f \in \text{filepools}} \frac{\mathit{Filpool\_Request\_Service\_Time}_f}{\text{SFSTIME}_f}$$

Filepool_Request_Service_Time is from the QUERY FILEPOOL STATUS command.

AVG ELIST SIZE (V). The average number of virtual machines in the eligible list queues (E0, E1, E2, E3) calculated by VMPRF.

This is taken from the average value of the <Elist> field in the VMPRF System Summary By Time report.

Avg Filepool Request Time (ms). The average time it takes for a request to the SFS file pool server machine to complete. This is calculated by:

$$\frac{\text{Agents In Call}}{\displaystyle\sum_{f \in \text{filepools}} \frac{\mathit{Total\_Filepool\_Requests}_f}{\text{SFSTIME}_f}}$$

Total_Filepool_Requests is from the QUERY FILEPOOL STATUS command.

AVG FIRST (T). The average response time in seconds for the first reply that returns to the screen. For non-fullscreen commands this is the command reflect on the screen. This is calculated by:

$$\frac{1}{\text{ETR (T)}} \times \sum_{t \in \text{TPNS machines}} \frac{\mathit{First\_Response}_t \times \mathit{Total\_Transmits}_t}{\mathit{TPNS\_Time}_t}$$

First_Response is the average first response given in the RSPRPT section of the TPNS reports. *Total_Transmits* is the total TPNS transmits and *TPNS_Time* is the run interval log time found in the Summary of Elapsed Time and Times Executed section of the TPNS reports.

AVG LAST (T). The average response time in seconds for the last response to the screen. If there is more than one TPNS this is calculated by:

$$\frac{1}{\text{ETR (T)}} \times \sum_{t \in \text{TPNS machines}} \frac{\mathit{Last\_Response}_t \times \mathit{Total\_Transmits}_t}{\mathit{TPNS\_Time}_t}$$

Last_Response is the average last response given in the RSPRPT section of the TPNS reports. *Total_Transmits* is the total TPNS transmits and

TPNS_Time is the run interval log time found in the Summary of Elapsed Time and Times Executed section of the TPNS reports.

AVG Lock Wait Time (ms). The average time it takes for an SFS lock conflict to be resolved. This is calculated by:

$$\frac{\sum_{f \in \text{filepools}} \frac{Lock\_Wait\_Time_f}{\text{SFSTIME}_f}}{\sum_{f \in \text{filepools}} \frac{Total\_Lock\_Conflicts_f}{\text{SFSTIME}_f}}$$

Lock_Wait_Time and *Total_Lock_Conflicts* are both from the QUERY FILEPOOL STATUS command.

AVG LUW Time (ms). The average duration of an SFS logical unit of work. This is calculated by:

$$\frac{\sum_{f \in \text{filepools}} \frac{Agent\_Holding\_Time_f}{\text{SFSTIME}_f}}{\sum_{f \in \text{filepools}} \frac{Begin\_LUWs_f}{\text{SFSTIME}_f}}$$

Agent_Holding_Time and *Begin_LUWs* are both from the QUERY FILEPOOL STATUS command.

AVG THINK (T). Average think time in seconds. The average think time determined by TPNS for all users. This is calculated by :

$$\frac{1}{\text{ETR (T)}} \times \sum_{t \in \text{TPNS machines}} \frac{Think\_Time_t \times Total\_Transmits_t}{TPNS\_Time_t}$$

Think_Time is the average think time given in the RSPRPT section of the TPNS reports. *Total_Transmits* is the total TPNS transmits and *TPNS_Time* is the run interval log time found in the Summary of Elapsed Time and Times Executed section of the TPNS reports.

Bactrian. A two-humped curve used to represent the think times for both active users and users who are logged on but inactive. The distribution includes those long think times that occur when a user is not actively issuing commands. Actual user data were collected and used as input to the creation of the Bactrian distribution.

BIO Request Time (ms). Average time required to process a block I/O request in milliseconds. This is calculated by:

$$\frac{\sum_{f \in \text{filepools}} \frac{Total\_BIO\_Request\_Time_f}{\text{SFSTIME}_f}}{\sum_{f \in \text{filepools}} \frac{Total\_BIO\_Requests_f}{\text{SFSTIME}_f}}$$

Total_BIO_Request_Time and *Total_BIO_Requests* are both from the QUERY FILEPOOL STATUS command.

Blocking Factor (Blocks/BIO). The average number of blocks read or written per Block I/O Request. This is calculated by:

$$\frac{\sum_{f \in \text{filepools}} \frac{Total\_DASD\_Block\_Transfers_f}{\text{SFSTIME}_f}}{\sum_{f \in \text{filepools}} \frac{Total\_BIO\_Requests_f}{\text{SFSTIME}_f}}$$

Total_DASD_Block_Transfers and *Total_BIO_Requests* are both from the QUERY FILEPOOL STATUS command.

Chaining Factor (Blocks/IO). The average number of blocks read or written per I/O request. This is calculated by:

$$\frac{\sum_{f \in \text{filepools}} \frac{Total\_DASD\_Block\_Transfers_f}{\text{SFSTIME}_f}}{\sum_{f \in \text{filepools}} \frac{Total\_IO\_Requests_f}{\text{SFSTIME}_f}}$$

Total_DASD_Block_Transfers and *Total_IO_Requests* are both from the QUERY FILEPOOL STATUS command.

Checkpoint. 1) In an SFS file pool server, the periodic processing that records a consistent state of the file pool on DASD. 2) In a CRR recovery server, the process used to maintain the log disks. All active syncpoint information is written to the logs.

Checkpoint Duration. The average time, in seconds, required to process an SFS checkpoint. This is calculated by:

$$\frac{1}{1000} \times \frac{\sum_{f \in \text{filepools}} Checkpoint\_Time_f}{\sum_{f \in \text{filepools}} Checkpoints\_Taken_f}$$

Checkpoint_Time and *Checkpoints_Taken* are from the QUERY FILEPOOL STATUS command.

Checkpoint Utilization. The percentage of time an SFS file pool server spends performing checkpoints. This is calculated by:

$$\frac{1}{10} \times \sum_{f \in \text{filepools}} \frac{Checkpoint\_Time_f}{\text{SFSTIME}_f}$$

Checkpoint_Time is from the QUERY FILEPOOL STATUS command.

Checkpoints Taken (delta). The number of checkpoints taken by all file pools on the system. This is calculated by:

$$\sum_{f \in \text{filepools}} Checkpoints\_Taken_f$$

Checkpoints_Taken is from the QUERY FILEPOOL STATUS command.

CMS BLOCKSIZE. The block size, in bytes, of the users′ CMS minidisks.

Command. In the context of reporting performance results, any user interaction with the system being measured.

CP/CMD. For the FS7F and IOB workloads this is the average amount of CP processor time used per command in milliseconds. For the PACE workload this is the average CP processor time per job in seconds. This is calculated by:

For the FS7F and IOB workloads:

$$10 \times \frac{\text{(TOTAL-TOTAL EMUL)}}{\text{ETR (T)}}$$

For PACE workload:

PBT/CMD-EMUL/CMD

CP/CMD (H). See CP/CMD. This is the hardware based measure. This is calculated by:

For 9221 Processors:
For FS7F and IOB workloads:

$$\frac{CP\_CPU\_PCT \times \text{TOTAL (H)}}{10 \times \text{ETR (T)}}$$

For PACE workload:

$$6000 \times \frac{CP\_CPU\_PCT \times \text{TOTAL (H)}}{\text{ETR (H)}}$$

CP_CPU_PCT is taken from the Host CPU Busy line in the CPU Busy/MIPs section of the RE0 report.

For all workloads running on 9121 and 9021 Processors:

PBT/CMD (H)-EMUL/CMD (H)

CP CPU/CMD (V) Server. CP processor time, in milliseconds, run in the designated server machine per command. This is calculated by:

$$\left(\frac{1}{V\_Time \times \text{ETR (T)}}\right) \times \sum_{s \in \text{server class}} (TCPU_s\text{-}VCPU_s)$$

TCPU is Total CPU busy seconds, *VCPU* is Virtual CPU seconds, and *V_Time* is the VMPRF time interval obtained from the Resource Utilization by User Class section of the VMPRF report.

DASD IO/CMD (V). The number of real SSCH or RSCH instructions issued to DASD, per job, used by the VSE guest in a PACE measurement. This is calculated by:

$$60 \times \frac{\text{DASD IO RATE (V)}}{\text{ETR (H)}}$$

DASD IO RATE (V). The number of real SSCH or RSCH instructions issued to DASD, per second, used by the VSE guest in a PACE measurement. This is calculated by:

$$\frac{\text{DASD IO TOTAL (V)}}{V\_Time}$$

V_Time is taken from the time stamps at the beginning of the VMPRF DASD Activity Ordered by Activity report.

DASD IO TOTAL (V). The number of real SSCH or RSCH instructions issued to DASD used by the VSE guest in a PACE measurement. This is calculated by:

$$\sum_{d \in \text{VSE Guest DASD}} Total_d$$

Total is taken from the Count column in the VMPRF DASD Activity Ordered by Activity report for the individual DASD volumes used by the VSE guest.

Deadlocks (delta). The total number of SFS file pool deadlocks that occurred during the measurement interval summed over all production file pools. A deadlock occurs when two users each request a resource that the other currently owns. This is calculated by:

$$\sum_{f \in \text{filepools}} Deadlocks_f$$

Deadlocks is from the QUERY FILEPOOL STATUS command.

DIAGNOSE. An instruction that is used to request CP services by a virtual machine. This instruction causes a SIE interception and returns control to CP.

DIAG 04/CMD. The number of DIAGNOSE code X′04′ instructions used per command. DIAGNOSE code X′04′ is the privilege class C and E CP function call to examine real storage. This is a product-sensitive programming interface. This is calculated by:

$$\frac{DIAG\_04}{\text{RTM\_Time} \times \text{ETR (T)}}$$

*DIAG*_04 is taken from the TOTALCNT column on the RTM PRIVOPS screen. RTM_Time is the total RTM time interval.

DIAG 08/CMD. The number of DIAGNOSE code X′08′ instructions used per command. DIAGNOSE code X′08′ is the CP function call to issue CP commands from an application. This is calculated by:

$$\frac{DIAG\_08}{\text{RTM\_Time} \times \text{ETR (T)}}$$

$DIAG\_08$ is taken from the TOTALCNT column on the RTM PRIVOPS screen. RTM_Time is the total RTM time interval.

DIAG 10/CMD. The number of DIAGNOSE code X′10′ instructions used per command. DIAGNOSE code X′10′ is the CP function call to release pages of virtual storage. This is calculated by:

$$\frac{DIAG\_10}{\text{RTM\_Time} \times \text{ETR (T)}}$$

$DIAG\_10$ is taken from the TOTALCNT column on the RTM PRIVOPS screen. RTM_Time is the total RTM time interval.

DIAG 14/CMD. The number of DIAGNOSE code X′14′ instructions used per command. DIAGNOSE code X′14′ is the CP function call to perform virtual spool I/O. This is calculated by:

$$\frac{DIAG\_14}{\text{RTM\_Time} \times \text{ETR (T)}}$$

$DIAG\_14$ is taken from the TOTALCNT column on the RTM PRIVOPS screen. RTM_Time is the total RTM time interval.

DIAG 58/CMD. The number of DIAGNOSE code X′58′ instructions used per command. DIAGNOSE code X′58′ is the CP function call that enables a virtual machine to communicate with 3270 virtual consoles. This is calculated by:

$$\frac{DIAG\_58}{\text{RTM\_Time} \times \text{ETR (T)}}$$

$DIAG\_58$ is taken from the TOTALCNT column on the RTM PRIVOPS screen. RTM_Time is the total RTM time interval.

DIAG 98/CMD. The number of DIAGNOSE code X′98′ instructions used per command. This allows a specified virtual machine to lock and unlock virtual pages and to run its own channel program. This is calculated by:

$$\frac{DIAG\_98}{\text{RTM\_Time} \times \text{ETR (T)}}$$

$DIAG\_98$ is taken from the TOTALCNT column on the RTM PRIVOPS screen. RTM_Time is the total RTM time interval.

DIAG 98/CMD (V) VTAM Servers. See DIAG 98/CMD for a description of this instruction. This represents the sum of all DIAGNOSE code X′98′ instructions per command for all VTAM and VSCS servers. This is calculated by:

$$\frac{DIAG\_98\_VTAM + DIAG\_98\_VSCS}{\text{ETR (T)}}$$

$DIAG\_98\_VTAM$ and $DIAG\_98\_VSCS$ are taken from the VMPRF Virtual Machine Communication by User Class report for the VTAM and VSCS server classes respectively.

DIAG A4/CMD. The number of DIAGNOSE code X′A4′ instructions used per command. DIAGNOSE code X′A4′ is the CP function call that supports synchronous I/O to supported DASD. This is calculated by:

$$\frac{DIAG\_A4}{\text{RTM\_Time} \times \text{ETR (T)}}$$

$DIAG\_A4$ is taken from the TOTALCNT column on the RTM PRIVOPS screen. RTM_Time is the total RTM time interval.

DIAG A8/CMD. The number of DIAGNOSE code X′A8′ instructions used per command. DIAGNOSE code X′A8′ is the CP function call that supports synchronous general I/O to fully supported devices. This is calculated by:

$$\frac{DIAG\_A8}{\text{RTM\_Time} \times \text{ETR (T)}}$$

$DIAG\_A8$ is taken from the TOTALCNT column on the RTM PRIVOPS screen. RTM_Time is the total RTM time interval.

DIAG 214/CMD. The number of DIAGNOSE code X′214′ instructions used per command. DIAGNOSE code X′214′ is used by the Pending Page Release function. This is calculated by:

$$\frac{DIAG\_214}{\text{RTM\_Time} \times \text{ETR (T)}}$$

$DIAG\_214$ is taken from the TOTALCNT column on the RTM PRIVOPS screen. RTM_Time is the total RTM time interval.

DIAG/CMD. The total number of DIAGNOSE instructions used per command or job. This is calculated by:

For the FS7F and IOB workloads:

$$\frac{1}{(\text{ETR (T)} \times RTM\_Time)} \times \sum_{x \in \text{DIAGNOSE}} TOTALCNT_x$$

For PACE workload:

$$\frac{60}{(\text{ETR (H)} \times RTM\_Time)} \times \sum_{x \in \text{DIAGNOSE}} TOTALCNT_x$$

$TOTALCNT$ is the count for the individual DIAGNOSE codes taken over the total RTM time interval on the RTM PRIVOPS Screen. RTM_Time is the total RTM time interval taken from the RTM PRIVOPS screen.

DISPATCH LIST. The average over time of the number of virtual machines (including loading virtual machines) in any of the dispatch list queues (Q0, Q1, Q2 and Q3).

$$\frac{1}{Num\_Entries} \times \sum_{t \in \text{SCLOG entries}} Q0_t + Q0L_t + Q1_t + Q1L_t + Q2_t + Q2L_t + Q3_t + Q3L_t$$

$Q0_t$, $Q0L_t$ .. are from the Q0CT, Q0L ... columns in the RTM SCLOG screen. $Num\_Entries$ is the total number of entries in the RTM SCLOG screen.

DPA (Dynamic Paging Area). The area of real storage used by CP to hold virtual machine pages, pageable CP modules and control blocks.

EDF. Enhanced Disk Format. This refers to the CMS minidisk file system.

Elapsed Time (H). The total time in seconds of a given measurement. This is the hardware estimate.

For 9221 Processors:

This is the Elapsed Time line in the RE0 file.

For 9121 and 9021 Processors this is calculated by:

$$Stop\_Time - Start\_Time$$

Stop_time and *Start_Time* are on the first line of the HISTDATA file.

ELIGIBLE LIST. The average over time of the number of virtual machines (including loading virtual machines) in any of the eligible list queues (E0, E1, E2 and E3).

$$\frac{1}{Num\_Entries} \times \sum_{t \in \text{SCLOG entries}} E0_t + E0L_t + E1_t + E1L_t + E2_t + E2L_t + E3_t + E3L_t$$

$E0_t$, $E0L_t$ .. are from the E0CT, E0L ... columns in the RTM SCLOG screen. *Num_Entries* is the total number of entries in the RTM SCLOG screen.

EMUL ITR. Emulation Internal Throughput Rate. The average number of transactions completed per second of emulation time.

This is from the EM_ITR field under TOTALITR of the RTM TRANSACT screen.

EMUL/CMD. For the FS7F and IOB workloads this is the amount of processor time spent in emulation mode per command in milliseconds. For the PACE workload this is the emulation processor time per job in seconds.

For the FS7F and IOB workloads this is calculated by:

$$10 \times \frac{\text{TOTAL EMUL}}{\text{ETR (T)}}$$

For the PACE workload this is calculated by:

$$6000 \times \frac{\text{TOTAL EMUL}}{\text{ETR (H)}}$$

EMUL/CMD (H). See EMUL/CMD. This is the hardware based measurement.

For the FS7F and IOB workloads this is calculated by:

$$10 \times \frac{\text{TOTAL EMUL (H)}}{\text{ETR (T)}}$$

For the PACE workload this is calculated by:

$$6000 \times \frac{\text{TOTAL EMUL (H)}}{\text{ETR (H)}}$$

ETR. External Throughput Rate. The number of commands completed per second, computed by RTM.

This is found in the NSEC column for ALL_TRANS for the total RTM interval time on the RTM Transaction screen.

ETR (H). The number of jobs completed per minute determined by the hardware instrumentation tools. It is calculated by:

$$60 \times \frac{Jobs}{\text{Elapsed Time (H)}}$$

Jobs is the number of jobs run in the workload. The values of *Jobs* are 28, 42 and 56 for the PACEX4, PACEX6 and PACEX8 workloads respectively.

ETR (T). See ETR. TPNS-based calculation of ETR. It is calculated by:

$$\sum_{t \in \text{TPNS machines}} \frac{Total\_Transmits_t}{TPNS\_Time_t}$$

Total_Transmits is found in the Summary of Elapsed Time and Times Executed section of TPNS report (TOTALS for XMITS by TPNS). *TPNS_Time* is the last time in requested (reduction) period minus the first time in requested (reduction) period. These times follow the Summary of Elapsed Time in the TPNS report.

ETR RATIO. This is the ratio of the RTM-based ETR calculation and the TPNS-based ETR calculation. This is calculated by :

$$\frac{\text{ETR}}{\text{ETR (T)}}$$

Expanded Storage. An optional integrated high-speed storage facility, available on certain processors, that allows for the rapid transfer of 4KB blocks between itself and real storage.

EXP. STORAGE. The amount of expanded storage used during a given run. See expanded storage.

External Response Time. The average response time, in seconds, for the last response to the screen. See AVG LAST (T).

FAST CLR/CMD. The number of fast path clears of real storage per command or job. This includes V=R and regular guests and is calculated by:

For the FS7F and IOB workloads:

$$\frac{Fast\_Clear\_Sec}{\text{ETR (T)}}$$

For PACE workload:

$$60 \times \frac{Fast\_Clear\_Sec}{\text{ETR (H)}}$$

Fast_Clear_Sec is taken from the NSEC column for the total RTM time interval for the FAST_CLR entry on the RTM SYSTEM screen.

File Pool. In SFS, a collection of minidisks managed by a server machine.

FP REQ/CMD (Q). Total file pool requests per command. This is calculated by:

$$\sum_{f \in filepools} \frac{Total\_Filepool\_Requests_f}{\text{SFSTIME}_f}$$

Total_Filepool_Requests is from the QUERY FILEPOOL STATUS command.

FREE TOTL/CMD. The number of requests for free storage per command or job. This includes V=R and regular guests and is calculated by:

For the FS7F and IOB workloads:

$$\frac{Free\_Total\_Sec}{\text{ETR (T)}}$$

For PACE workload:

$$60 \times \frac{Free\_Total\_Sec}{\text{ETR (H)}}$$

Free_Total_Sec is taken from the NSEC column for the total RTM time interval on the RTM SYSTEM screen.

FREE UTIL. The proportion of the amount of available free storage actually used. This is calculated by :

$$\frac{Free\_Size}{\text{FREEPGS} \times 4096}$$

Free_Size is found in the FREE column for the total RTM time interval (<-..) on the RTM SYSTEM screen.

FREEPGS. The total number of pages used for FREE storage (CP control blocks).

This is found in the FPGS column for the total RTM time interval (<-..) on the RTM SYSTEM screen.

FST (File Status Table). CMS control block that contains information about a file belonging to a minidisk or SFS directory.

GB. Gigabytes. 1024 megabytes.

GRAF-ID. Display terminal address.

GUEST SETTING. This field represents the type of VSE guest virtual machine in a PACE measurement. This fields possible values are V=V, V=F or V=R.

GUESTWT/CMD. The number of entries into guest enabled wait state per job. This is calculated by:

$$60 \times \frac{\text{GUESTWT/SEC}}{\text{ETR (H)}}$$

GUESTWT/SEC. The number of entries into guest enabled wait state per second.

This field is taken from the NSEC column for the RTM total count since last reset, for the GUESTWT field in the RTM SYSTEM screen.

Hardware Instrumentation.. See Processor Instrumentation

HT5. One of the CMS-intensive workloads used in the Large Systems Performance Reference (LSPR) to evaluate relative processor performance.

IML MODE. This is the hardware IML mode used in VSE guest measurements. The possible values for this field are 370, ESA, or LPAR.

Instruction Path Length. The number of machine instructions used to run a given command, function or piece of code.

Internal Response Time. The response time as seen by CP. This does not include line or terminal delays.

IO TIME/CMD (Q). Total elapsed time in seconds spent doing SFS file I/Os per command. This is calculated by:

$$\frac{1}{(1000 \times \text{ETR (T)})} \times \sum_{(f \in filepools)} \frac{Total\_BIO\_Request\_Time_f}{\text{SFSTIME}_f}$$

Total_BIO_Request_Time is from the QUERY FILEPOOL STATUS command.

IO/CMD (Q). SFS file I/Os per command. This is calculated by:

$$\frac{1}{\text{ETR (T)}} \times \sum_{f \in filepools} \frac{Total\_IO\_Requests_f}{\text{SFSTIME}_f}$$

Total_IO_Requests is from the QUERY FILEPOOL STATUS command.

ITR. Internal Throughput Rate. This is the number of units of work accomplished per unit of processor busy time in an nonconstrained environment. For the FS7F and IOB workloads this is represented as commands per processor second. For the PACE workload this is represented as jobs per processor minute. This is calculated by:

For FS7F and IOB workloads:

This is found from the TOTALITR for SYS_ITR on the RTM TRANSACT screen.

For PACE workload:

$$100 \times \frac{\text{ETR (H)}}{\text{UTIL/PROC}}$$

ITR (H). See ITR. This is the hardware based measure. In this case, ITR is measured in external commands per unit of processor busy time. For the FS7F and IOB workloads this is represented as commands per processor second, while for the PACE workload this is represented in jobs per processor minute. This is calculated by:

For the FS7F and IOB workloads:

$$100 \times \frac{\text{ETR (T)}}{\text{TOTAL (H)}}$$

For the PACE workloads:

$$6000 \times \frac{Jobs}{\text{Elapsed time (H)} \times \text{TOTAL (H)}}$$

Jobs is the number of jobs run in the workload. The values of *Jobs* are 7, 28, 42 and 56 for the PACE1, PACEX4, PACEX6 and PACEX8 workloads respectively.

ITRR. Internal Throughput Rate Ratio. This is the RTM based ITR normalized to a specific run. This is calculated by:

$$\frac{\text{ITR}}{\text{ITR}_1}$$

ITR_1 is the ITR of the first run in a given table.

ITRR (H). See ITRR. This is the ITR (H) normalized to a specific run. This is calculated by:

$$\frac{\text{ITR (H)}}{\text{ITR (H)}_1}$$

ITR (H)_1 is the ITR (H) of the first run in a given table.

Inter-user Communication Vehicle (IUCV). A VM generalized CP interface that helps the transfer of messages either among virtual machines or between CP and a virtual machine.

k. Multiple of 1000.

Kb. Kilobits. One kilobit is 1024 bits.

KB. Kilobytes. One kilobyte is 1024 bytes.

LUW Rollbacks (delta). The total number of SFS logical units of work that were backed out during the measurement interval, summed over all production file pools. This is calculated by:

$$\sum_{f \in \text{filepools}} LUW\_Rollbacks_f$$

LUW_Rollbacks is from the QUERY FILEPOOL STATUS command.

MASTER EMUL. Total emulation state utilization for the master processor. For uniprocessors this is the same as TOTAL EMUL and is generally not shown.

This is taken from the %EM column for the first processor listed in the LOGICAL CPU STATISTICS section of the RTM CPU screen. The total RTM interval time value is used (<-..).

MASTER EMUL (H). Total emulation state utilization for the master processor. For uniprocessors this is the same as TOTAL EMUL and is generally not shown. This is the hardware based calculation.

This is taken from the %CPU column of the GUES-CPn line of the REPORT file for the master processor number as shown by RTM. In RTM, the first processor listed on the CPU screen is the master processor.

MASTER TOTAL. Total utilization of the master processor. For uniprocessor this is the same as TOTAL and is generally not shown.

This is taken from the %CPU column for the first processor listed in the LOGICAL CPU STATISTICS section of the RTM CPU screen. The total RTM interval time value is used (<-..).

MASTER TOTAL (H). Total utilization of the master processor. For uniprocessor this is the same as TOTAL (H) and is generally not shown. This is the hardware based calculation.

This is taken from the %CPU column of the SYST-CPn line of the REPORT file for the master processor number as shown by RTM. In RTM, the first processor listed on the CPU screen is the master processor.

MB. Megabytes. One megabyte is 1,048,576 bytes.

MDC Hit Ratio. Minidisk Cache Hit Ratio. The ratio of requested blocks that are found in expanded storage for a read operation to the total number of blocks read.

This is the from the RTM MDHR field for the total RTM time interval (<-..) on the RTM SYSTEM screen.

MDC Mods. Minidisk Cache Modifications. The number of times per second blocks were written in the cache, excluding the writes that occurred as a result of minidisk cache misses.

This is taken from the NSEC column for the RTM MDC_MO field for the total RTM time interval on the RTM SYSTEM screen.

MDC Reads. Minidisk Cache Reads. The number of times per second blocks were found in the cache as the result of a read operation.

This is taken from the NSEC column for the RTM MDC_HT field for the total RTM time interval on the RTM SYSTEM screen.

MDC Writes. Minidisk Cache Writes. The number of CMS Blocks moved per second from main storage to expanded storage.

This is taken from the NSEC column for the RTM MDC_PW field for the total RTM time interval on the RTM SYSTEM screen.

Millisecond. One one-thousandth of a second.

Minidisk Caching. Refers to the use of a portion of expanded storage as a write-through, (data written to minidisks is also written to the cache), minidisk cache for CMS 4KB block size data. It is used to help eliminate I/O bottlenecks and improve system response time by replacing actual DASD I/Os with faster expanded storage I/O.

Minidisk File Cache. A buffer used by CMS when a file is read or written to sequentially. When a file is read sequentially, CMS reads ahead as many blocks as will fit into the cache. When a file is written sequentially, completed blocks are accumulated until the cache is filled and then are written out together.

MPG. Multiple preferred guests is a facility on a processor that has the Processor Resource/Systems Manager* (PR/SM) feature installed. This facility supports up to 6 preferred virtual machines. One can be a V=R, the others being V=F.

ms. Millisecond.

Non-shared Storage. The portion of a virtual machine's storage that is unique to that virtual machine, (as opposed to shared storage such as a saved segment that is shared among virtual machines). This is usually represented in pages.

NONPAGE RIO/CMD (V). The number of real SSCH and RSCH instructions issued per command for purposes other than paging. This is calculated by:

$$\text{RIO/CMD (V)} - \text{PAGE IO/CMD (V)}$$

NONTRIV INT. Non-trivial Internal response time in seconds. The number of transactions that completed with more than one drop from Q1 or one or more drops from Q0, Q2, or Q3 per second.

This is from TOTALTTM for the RTM NTRIV field on the RTM TRANSACT screen.

NUCLEUS SIZE (V). The resident CP nucleus size in kilobytes.

This is from the <K bytes> column on the Total Resident Nucleus line in the VMPRF System Configuration Report.

PAGE/CMD. The number of pages moved between real storage and DASD per command or job. This is calculated by:

For the FS7F and IOB workloads:

$$\frac{\text{READS/SEC + WRITES/SEC}}{\text{ETR (T))}}$$

For PACE workload:

$$60 \times \frac{\text{READS/SEC + WRITES/SEC}}{\text{ETR (H)}}$$

PAGE IO RATE (V). The number of real SSCH or RSCH instructions issued on behalf of system paging.

This is the sum of all the entries in the SSCH+RSCH column for Page devices listed in the VMPRF DASD System Areas by Type report.

PAGE IO/CMD (V). The number of real SSCH and RSCH instructions issued per command on behalf of system paging. This is calculated by:

$$\frac{\text{PAGE IO RATE (V)}}{\text{ETR (T)}}$$

Path length. See Instruction Path Length

PBT/CMD. For the FS7F and IOB workloads, this is the number of milliseconds of processor activity per command. For the PACE workload, this is the number of seconds of processor activity per job. This is calculated by:

For the FS7F and IOB workloads:

$$10 \times \frac{\text{TOTAL}}{\text{ETR (T)}}$$

For the PACE workload:

$$6000 \times \frac{\text{TOTAL}}{\text{ETR (H)}}$$

PBT/CMD (H). See PBT/CMD. This is the hardware based measure.

For the FS7F and IOB workloads:

$$10 \times \frac{\text{TOTAL (H)}}{\text{ETR (T)}}$$

For the PACE workload:

$$6000 \times \frac{\text{TOTAL (H)}}{\text{ETR (H)}}$$

PD4. One of the CMS-intensive workloads used in the Large Systems Performance Reference (LSPR) to evaluate relative processor performance.

PGBLPGS. The number of system pageable pages available.

This is from the PPAG field for the total RTM time interval (<-) on the RTM SYSTEM screen.

PGBLPGS/USER. The number of system pageable pages available per user. This is calculated by:

$$\frac{\text{PGBLPGS}}{\text{USERS}}$$

Privileged Operation. Any instruction that must be run in supervisor state.

PRIVOP/CMD. The number of virtual machine privileged instructions simulated per command or job. This does not include DIAGNOSE instructions. This is calculated by:

For FS7F and IOB workloads:

$$\frac{1}{(\text{ETR (T)}) \times RTM\_Time)} \times \sum_{x \in \text{privops}} TOTALCNT_x$$

For PACE workload:

$$\frac{60}{(\text{ETR (H)} \times RTM\_Time)} \times \sum_{x \in \text{privops}} TOTALCNT_x$$

$TOTALCNT$ is the count for the individual privop taken over the total RTM time interval on the RTM PRIVOPS Screen. RTM_Time is the total RTM time interval taken from the RTM PRIVOPS screen. **Note**: PRIVOPS are recorded differently in 370 and XA modes.

PRIVOPS (Privileged Operations). See Privileged Operation.

Processor Instrumentation. An IBM internal tool used to obtain hardware-related data such as processor utilizations.

Processor Utilization. The percent of time that a processor is not idle.

PROCESSORS. The data field denoting the number of processors that were active during a measurement.

This is from the NC field under CPU statistics on the RTM CPU screen.

Production File Pool. An SFS file pool in which users are enrolled with space. All SFS read/write activity is to production file pools.

QUICKDSP ON. When a virtual machine is assigned this option, it bypasses the normal scheduler algorithm and is placed on the dispatch list immediately when it has work to do. It does not spend time in the eligible lists. QUICKDSP can be specified either via a CP command or in the CP directory entry.

READS/SEC. The number of pages read per second done for system paging.

This is taken from the NSEC column for the PAGREAD field for the total RTM time interval on the RTM SYSTEM screen.

REAL STORAGE. The amount of real storage used for a particular measurement.

Relative Share. A relative share allocates to a virtual machine a portion of the total system resources minus those resources allocated to virtual machines with an ABSOLUTE share. A virtual machine with a RELATIVE share receives access to system resources that is proportional with respect to other virtual machines with RELATIVE shares.

RESERVE. See SET RESERVED

RIO/CMD (V). The number of real SSCH and RSCH instructions issued per command. This is calculated by:

For FS7F and IOB workloads:

$$\frac{\text{RIO RATE (V)}}{\text{ETR (T)}}$$

For PACE workload:

$$60 \times \frac{\text{RIO RATE (V)}}{\text{ETR (H)}}$$

RIO RATE (V). The number of real SSCH and RSCH instructions issued per second.

This is taken from the I/O Rate column for the overall average on the VMPRF System Performance Summary by Time report; the value reported does not include assisted I/Os.

Rollback Requests (delta). The total number of SFS rollback requests made during a measurement. This is calculated by:

$$\sum_{f \in \text{filepools}} Rollback\_Requests_f$$

Rollback_Requests is from the QUERY FILEPOOL STATUS command.

Rollbacks Due to Deadlock (delta). The total number of LUW rollbacks due to deadlock that occurred during the measurement interval over all production file pools. A rollback occurs whenever a deadlock condition cannot be resolved by the SFS server. This is calculated by:

$$\sum_{f \in \text{filepools}} Rollbacks\_Due\_to\_Deadlock_f$$

Rollbacks_Due_to_Deadlock is from the QUERY FILEPOOL STATUS command.

RTM. Realtime Monitor. A licensed program realtime monitor and diagnostic tool for performance monitoring, analysis, and problem solving.

RTM/ESA. See RTM.

Run ID. An internal use only name used to identify a performance measurement.

SAC Calls / FP Request. The average number of calls within the SFS server to its Storage Access Component (SAC) per file pool request. In environments where there are multiple file pools, this average is taken over all file pool servers. This is calculated by:

$$\frac{\sum_{f \in \text{filepools}} \frac{Sac\_Calls_f}{\text{SFSTIME}_f}}{\sum_{f \in \text{filepools}} \frac{Total\_Filepool\_Requests_f}{\text{SFSTIME}_f}}$$

Sac_Calls and *Total_Filepool_Requests* are from the QUERY FILEPOOL STATUS command.

Seconds Between Checkpoints. The average number of seconds between SFS file pool checkpoints in the average file pool. This is calculated by:

$$\frac{1}{\sum_{f \in \text{filepools}} \frac{Checkpoints\_Taken_f}{\text{SFSTIME}_f}}$$

Checkpoints_Taken is from the QUERY FILEPOOL STATUS command.

SET RESERVED (Option). This is a CP command that can be used to allow a V=V virtual machine to have a specified minimum number of pages resident in real storage. It is used to reduce paging and improve performance for a given virtual machine.

SFSTIME. The elapsed time in seconds between QUERY FILEPOOL STATUS invocations for a given file pool done at the beginning and end of a measurement.

SFS TIME/CMD (Q). Total elapsed time per command, in seconds, required to process SFS server requests. This is calculated by:

$$\frac{1}{\text{ETR (T)}} \times \sum_{f \in \text{filepools}} \frac{Filepool\_Request\_Service\_Time_f}{\text{SFSTIME}_f}$$

Filepool_Request_Service_Time is from the QUERY FILEPOOL STATUS command.

SHARE. The virtual machine's SHARE setting. The SET SHARE command and the SHARE directory statement allow control of the percentage of system resources a virtual machine receives. These resources include processors, real storage and paging I/O capability. A virtual machine receives its proportion of these resources according to its SHARE setting. See Relative and Absolute Share.

Shared Storage. The portion of a virtual machines storage that is shared among other virtual machines (such as saved segments). This is usually represented in pages.

SHRPGS. The number of shared frames currently resident.

SIE. ESA Architecture instruction to Start Interpretive Execution. This instruction is used to run a virtual machine in emulation mode.

SIE INTCPT/CMD. The number of exits from SIE which are SIE interceptions per command or job. SIE is exited either by interception or interruption. An intercept is caused by any condition that requires CP interaction such as I/O or an instruction that has to be simulated by CP. This is calculated by:

$$\frac{Percent\_Intercept \times \text{SIE/CMD}}{100}$$

Percent_Intercept is taken from the %SC field for average of all processors for the total RTM time interval (<-..) on the RTM CPU screen.

SIE/CMD. SIE instructions used per command or job. This is calculated by:

For the FS7F and IOB workloads:

$$\frac{SIE\_SEC}{\text{ETR (T)}}$$

For PACE workload:

$$60 \times \frac{SIE\_SEC}{\text{ETR (H)}}$$

SIE_SEC is taken from the XSI field for the total for all processors for the total RTM time interval (<-..) on the RTM CPU screen.

T/V Ratio. See TVR

TOT CPU/CMD (V) Server. The total amount of processor time, in milliseconds, for the server virtual machine(s). This is calculated by:

$$\frac{1}{(V\_Time \times \text{ETR (T)})} \times \sum_{s \in \text{server class}} Total\_CPU\_Secs_s$$

Total_CPU_Secs and *V_Time* are from the Resource Utilization by User Class section of the VMPRF reports.

TOT INT. Total Internal Response Time in seconds. Internal response time averaged over all trivial and non-trivial transactions.

This is the value for TOTALTTM for ALL_TRANS on the RTM TRANSACT screen.

TOT INT ADJ. Total internal response time (TOT INT) reported by RTM, adjusted to reflect what the response time would have been had CP seen the actual command rate (as recorded by TPNS). This is a more accurate measure of internal response time than TOT INT. In addition, TOT INT ADJ can be directly compared to external response time (AVG LAST (T)) as they are both based on the same, TPNS-based measure of command rate. This is calculated by:

$$\text{TOT INT} \times \text{ETR RATIO}$$

TOTAL. The total processor utilization for a given measurement summed over all processors.

This comes from the %CPU column for all processors for the total RTM interval time (<-..) on the RTM CPU screen.

TOTAL (H). See TOTAL. This is the hardware based measurement.

For 9221 Processors:

This is taken from the Total CPU Busy line in the CPU Busy/Mips section of the RE0 report.

For 9121 and 9021 Processors, this is calculated by:

$$\text{UTIL/PROC (H)} \times \text{PROCESSORS}$$

TOTAL EMUL. The total emulation state time for all users across all online processors. This indicates the percentage of time the processors are in emulation state.

This comes from the %EM column for all processors for the total RTM interval time (<-..) on the RTM CPU screen.

TOTAL EMUL (H). The total emulation state time for all users across all online processors. This indicates the percentage of time the processors are in emulation state. This is calculated by:

For 9221 Processors:

This comes from the SIE CPU Busy / Total CPU Busy (PCT) line in the RE0 report.

For 9121 and 9021 Processors:

The %CPU column for the GUES-ALL line of the REPORT file times the number of processors.

TPNS. Teleprocessing Network Simulator. A licensed program terminal and network simulation tool that provides system performance and response time information.

Transaction. A user/system interaction as counted by CP. For a single-user virtual machine a transaction should roughly correspond to a command. It does not include network or transmission delays and may include false transactions. False transactions can be those that wait for an external event, causing them to be counted as multiple transactions, or those that process more than one command without dropping from queue, causing multiple transactions to be counted as one.

TRACE TABLE (V). The size in kilobytes of the CP trace table.

This is the value of the <K bytes> column on the Trace Table line in the VMPRF System Configuration Report.

Transaction (T). This is the interval from the time the command is issued until the last receive prior to the next send. This includes clear screens as a result of an intervening MORE... or HOLDING condition.

TRIV INT. Trivial Internal Response Time in seconds. The rate per second for transactions that complete with one and only one drop from Q1 and no drops from Q0, Q2, and Q3.

This is from TOTALTTM for the TRIV field on the RTM TRANSACT screen.

TVR. Total to Virtual Ratio. This is the ratio of total processor utilization to virtual processor utilization. This is calculated by:

$$\frac{\text{TOTAL}}{\text{TOTAL EMUL}}$$

TVR (H). See TVR. Total to Virtual Ratio measured by the hardware monitor. This is calculated by:

$$\frac{\text{TOTAL (H)}}{\text{TOTAL EMUL (H)}}$$

USERS. The number of virtual machines logged on to the system during a measurement interval that are associated with simulated end users. This includes active and inactive virtual machines but does not include service machines.

UTIL/PROC. Per processor utilization. This is calculated by:

$$\frac{\text{TOTAL}}{\text{PROCESSORS}}$$

UTIL/PROC (H). Per processor utilization reported by the hardware.

For 9221 Processors, this is calculated by:

$$\frac{\text{TOTAL (H)}}{\text{PROCESSORS}}$$

For 9121 and 9021 Processors:

This is taken from the %CPU column in the SYST-ALL line of the REPORT file.

VIO RATE. The total number of all virtual I/O requests per second for all users in the system.

This is from the ISEC field for the total RTM time interval (<-) on the RTM SYSTEM screen.

VIO/CMD. The average number of virtual I/O requests per command or job for all users in the system. This is calculated by:

For the FS7F and IOB workloads:

$$\frac{\text{VIO RATE}}{\text{ETR (T))}}$$

For PACE workload:

$$60 \times \frac{\text{VIO RATE}}{\text{ETR (H)}}$$

VIRT CPU/CMD (V) Server. Virtual processor time, in milliseconds, run in the designated server(s) machine per command. This is calculated by:

$$\frac{1}{(V\_Time \times \text{ETR (T)})} \times \sum_{s \in \text{server class}} Virt\_CPU\_Secs_s$$

Virt_CPU_Secs and *V_Time* are from the Resource Utilization by User Class section of the VMPRF reports.

VM MODE. This field is the virtual machine setting (370, XA or ESA) of the VSE guest virtual machine in PACE measurements.

VM SIZE. This field is the virtual machine storage size of the VSE guest virtual machine in PACE measurements.

VMPAF. Virtual Machine Performance Analysis Facility. A tool used for performance analysis of VM systems.

VMPRF. VM Performance Reporting Facility. A licensed program that produces performance reports and history files from VM/XA or VM/ESA monitor data.

VSCSs. The number of virtual machines running VSCS external to VTAM during a measurement interval.

VSE SUPERVISOR. This field is the VSE supervisor mode used in a PACE measurement. This field's possible values are 370, ESA, and VMESA.

VSE WKSET (V). The average working set size in pages of the VSE guest machine in the PACE workload.

This is found in the WSS column in the VMPRF Resource Utilization by User Class report for the VSE user class.

VTAMs. The number of virtual machines running VTAM during a measurement interval.

V=F. Virtual equals fixed machine. A virtual machine that has a fixed, contiguous area of real storage. Unlike V=R, storage does not begin at page 0. For guests running V=F, CP does not page this area. Requires the PR/SM hardware feature to be installed.

V=R. Virtual equals real machine. Virtual machine that has fixed, contiguous area of real storage starting at page 0. CP does not page this area.

V=V. Virtual equals virtual machine. Default storage processing. CP pages the storage of a V=V machine in and out of real storage.

WKSET (V). The average working set size. This is the scheduler's estimate of the amount of storage the average user will require, in pages.

This is the average of the values for WSS in the VMPRF Resource Utilization by User report, (found in the Sum/Avg line).

WKSET (V) Server. Total working set of a related group of server virtual machine(s). This is calculated by:

$$\sum_{s \in \text{server Logged Users}} Avg\_WSS_s$$

Avg_WSS is found in the Avg WSS column in the VMPRF Resource Utilization by User Class report for each class of server.

WRITES/SEC. The number of page writes per second done for system paging.

This is taken from the NSEC column for the PAWRIT field for the total RTM time interval on the RTM SYSTEM screen.

XSTOR IN/SEC. The number of pages per second read into main storage from expanded storage. This includes fastpath and non-fastpath pages and is calculated by:

$$Fastpath\_In + NonFastpath\_In$$

Fastpath_In and *NonFastpath_In* are taken from the NSEC column for the XST_PGIF and XST_PGIS fields for the total RTM time interval on the RTM SYSTEM screen.

XSTOR OUT/SEC. The number of pages per second written from main storage into expanded storage. This is taken from the NSEC column for the XST_PGO field for the total RTM time interval on the RTM SYSTEM screen.

XSTOR/CMD. The number of pages read into main storage from expanded storage and written to expanded storage from main storage per command or job. This is calculated by:

For the FS7F and IOB workloads:

$$\frac{\text{XSTOR IN/SEC + XSTOR OUT/SEC}}{\text{ETR (T)}}$$

For PACE workload:

$$60 \times \frac{\text{XSTOR IN/SEC + XSTOR OUT/SEC}}{\text{ETR (H)}}$$

Table Definitions

id	File	Page	References
I52118	M742TABL	26	26, 26, 26, 26, 27, 27, 27, 27, 27, 28
I52119	M742TABL	26	26
I57242	M742TABC	29	30, 30, 30, 30, 30, 31, 31, 31, 31, 31
I57243	M742TABC	29	29
I32489	M742SFST	33	33, 33, 34, 34, 34, 34, 34, 35, 35, 35, 35
I32490	M742SFST	33	33
I33509	M480SFST	37	37, 37, 37, 38, 38, 38, 38, 38, 39, 39, 39
I33510	M480SFST	37	37
I33717	M170TABL	41	42, 42, 42, 42, 42, 42, 43, 43, 43, 43
I33718	M170TABL	41	41
I46908	MOV742	47	47, 47, 47, 48, 48, 48, 48, 48, 49, 49
I46909	MOV742	47	47
I35145	MOV480	51	51, 51, 51, 52, 52, 52, 52, 52, 53, 53
I35146	MOV480	51	51
I23102	M4VVVE	57	57, 57, 57, 57, 57, 57, 57
I23103	M4VVVE	57	57
I23172	M4VVVV	58	58, 58, 58, 58, 58, 58, 58
I23173	M4VVVV	58	58
I22843	M4VVRE	59	59, 59, 59, 59, 59, 59, 59
I22845	M4VVRE	59	59
I45155	FPMDSUMM	60	61, 61, 61, 61, 61, 61, 61
I45156	FPMDSUMM	60	60
MS3	MMIGSUM3	68	69
I30523	VSELOC1A	84	84, 84, 84, 84, 84, 84, 85
I30524	VSELOC1A	84	84
I30481	VSELOC2A	86	86, 86, 86, 86, 86, 86, 87
I30482	VSELOC2A	86	86
I47745	LOC1SUMM	90	90, 90, 90, 90, 90, 90, 91
I47746	LOC1SUMM	90	90
I30844	SINGSUM1	93	94, 94, 94, 94, 94, 94, 95
I30845	SINGSUM1	93	93
I30852	SINGSUM2	96	96, 96, 96, 96, 96, 96, 96, 97
I27763	CMSTTAB1	101	101, 101, 101, 101, 102, 102, 102, 102, 102, 103
I27764	CMSTTAB1	101	101
I28246	CMSTTAB2	105	105, 105, 105, 106, 106, 106, 106, 106, 107, 107
I28247	CMSTTAB2	105	105
I28305	CMSTTAB3	108	109, 109, 109, 109, 109, 110, 110, 110, 110, 110
I28306	CMSTTAB3	108	108

I28136	CMSTTAB4	112	112, 112, 113, 113, 113, 113, 113, 114, 114, 114
I28137	CMSTTAB4	112	112
I28187	CMSTTAB5	115	115, 115, 115, 115, 116, 116, 116, 116, 116, 117
I28188	CMSTTAB5	115	115
I57229	NULWPSW	120	120, 120, 120, 120, 120, 120, 121
I57230	NULWPSW	120	120
I34026	CMSCACH	123	124, 124, 124, 124, 124, 125, 125, 125, 125, 125
I34027	CMSCACH	123	123
I40835	PCAPTABL	133	133, 133, 133, 133, 133, 134, 134, 134, 134, 134, 134
I40836	PCAPTABL	133	133
I45315	3745MTAB	139	139, 139, 140, 140, 140, 140, 140, 141, 141, 141, 141
I45316	3745MTAB	139	139
I36277	MONCOST	143	144, 144, 144, 144, 144, 145, 145, 145, 145, 145
I36278	MONCOST	143	143
I59216	CPARBTAB	148	148, 148, 149, 149, 149, 149, 149, 150, 150, 150
I59217	CPARBTAB	148	148
I50714	HLASM	152	153, 153, 153, 153, 153, 154, 154, 154, 154, 154
I50715	HLASM	152	152
TIPLST	REXXTIP	158	158, 158
FIXTBL	REXXTIP	158	158
REXXTB	REXXTIP	160	160, 160, 160, 161, 161, 162, 162, 162, 162, 163, 163, 163, 164, 164, 164, 165, 165, 165, 165, 165, 165, 166, 166, 166, 166, 167, 167, 167, 168, 168, 169, 169, 170, 170, 171, 171, 171, 171, 172, 172, 172, 172, 172, 173, 173, 173, 173, 174, 174, 175, 175, 175, 175, 176, 176, 176, 177, 178
REXXT2	REXXTIP	160	161, 164
I56542	VARTAB1	181	182, 182, 182, 182, 182, 183, 183, 183, 183, 183
I56543	VARTAB1	181	181
I56544	VARTAB2	184	184, 184, 184, 184, 184, 184, 184, 185, 185, 185
I56545	VARTAB2	184	184
I31338	ASFSCNT	189	189, 189, 189, 189, 189, 189, 189, 189, 189, 190, 190, 190, 190, 190
I31339	ASFSCNT	189	189, 190

Figures

id	File	Page	References
CMSMIGR	MMIGCMS	23	1
CMSMIGI	MMIGCMS	24	2
OVITR	MOVMIGR	44	3
OVRESP	MOVMIGR	45	4
4VSEPAC	M4VSE	54	5
4VSEPAE	M4VSE	55	6
L21FPMD	M4VFPMD	60	7
VSELOC1	VSELOC1	82	8
VSELOC2	VSELOC1	83	9
21V170E	VSETEMP	88	10
21V170I	VSETEMP	89	11
PACEX1V	VSESING	92	12
PACEXVD	VSESING	93	13
RL21SM1	RL21SUM	135	15
RL21SM2	RL21SUM	136	16
QRYLSEG	ALSEG	213	18 210

Headings

id	File	Page	References
R21NOTI	REPORT21 SCRIPT	1	Notices
R21PI	REPORT21 SCRIPT	1	Programming Information
R21TRAD	REPORT21 SCRIPT	2	Trademarks
R21ACK	REPORT21 SCRIPT	3	Acknowledgements
R21ABS	REPORT21 SCRIPT	4	Abstract
APUBS	REPORT21 SCRIPT	5	Referenced Publications
R21KEY	REPORT21 SCRIPT	6	Summary of Key Findings
GCHANG	REPORT21 SCRIPT	9	Changes That Affect Performance 23
GIMPROV	REPORT21 SCRIPT	10	Performance Improvements 6, 9, 47, 51
VDISK	GIMPROV	10	Virtual Disk in Storage
LOAD	GIMPROV	10	Load Wait State PSW
IUCV	GIMPROV	11	IUCV Processor Use
CCWFP	GIMPROV	11	Fast Path CCW Translation
SHARSEG	GIMPROV	11	Shared Segment Serialization
DATACOM	GIMPROV	11	Support for Data Compression
EXECUTE	GIMPROV	12	EXECUTE XEDIT Macro
CACHE	GIMPROV		

59 9121-320 / V=R MODE=ESA

M4VFPMD REPORT21 SCRIPT
60 9221-170 / V=R MODE=ESA with Full Pack Minidisks
6, 11

MSESMIG REPORT21 SCRIPT
62 VMSES/E
6

MMIGSUM REPORT21 SCRIPT
68 Migration from Other VM Releases
7

MMIGMAT MMIGSUM
68 Migration Performance Measurements Matrix

MMIGCM2 MMIGSUM
70 Migration Summary: CMS-Intensive Environment
68, 68

NUVDISK REPORT21 SCRIPT
76 Virtual Disk in Storage
7, 10

VSEVDSK REPORT21 SCRIPT
80 VSE Guest

VSELOC1 REPORT21 SCRIPT
82 9121-320 / VSE Lock File and Temporary Files

VSETEMP REPORT21 SCRIPT
88 9221-170 / Progressive Use by VSE Guests

VSESING REPORT21 SCRIPT
92 9221-170 / Single Thread PACEX1

CMSTDSK REPORT21 SCRIPT
98 CMS Environment

MNEW REPORT21 SCRIPT
118 Additional Functional Enhancements

NULWPSW REPORT21 SCRIPT
119 Load Wait State PSW Improvements
6, 118

CMSCACH REPORT21 SCRIPT
122 Adjusting the Minidisk File Cache Size
7, 12, 118

REXXL2 REPORT21 SCRIPT
126 REXX SAA Level 2 Architecture
7, 118

ADDEVAL REPORT21 SCRIPT
127 Additional Evaluations

PROCCAP REPORT21 SCRIPT
128 Processor Capacity
7, 127

RL21SUM REPORT21 SCRIPT
135 VSE/ESA Guest Performance
7, 127

3745M REPORT21 SCRIPT
137 3745 Comparison to CTCA
7, 127

MONCOST REPORT21 SCRIPT
142 CP Monitor Overhead
8, 127

CPARB REPORT21 SCRIPT
146 Minidisk Cache Tuning: Restricting the Arbiter
8, 26, 127, 131

HLASM REPORT21 SCRIPT
151 High Level Assembler Evaluation
8, 127

REXXTIP REPORT21 SCRIPT
155 VM/ESA REXX Performance Guidelines
8, 12, 127

REXX01 REXXTIP
155 REXX Environments
158

REXX11 REXXTIP
156 XEDIT search order

REXX26 REXXTIP
157 REXX Scanning and Executing
174, 178, 179

REXXSM REXXTIP
157 Summary of REXX Performance Tips

REXX06 REXXTIP
160 Invoking Commands From an Exec
158, 158

REXX07 REXXTIP
160 To Invoke a CP Command: address ′COMMAND′ ′CP cpcmd′
158, 158

REXX08 REXXTIP

160 To Invoke a CMS command: address ′COMMAND′ ′cmscmd′
158, 158

REXX09 REXXTIP

160 To Invoke an Exec: address ′COMMAND′ ′EXEC exec′
158, 158

REXX80 REXXTIP

161 Additional CMS Short Cuts
158, 158

REXX24 REXXTIP

161 Abbreviate CMS Command Parameters (But Not CMS Commands)
158, 158

REXX69 REXXTIP

161 Use Variable Record Format Files (RECFM=V)
158, 158

REXX45 REXXTIP

161 Use the STEM Option of EXECIO
158, 158

REXX46 REXXTIP

162 With EXECIO Loop, Use Single FINIS
158, 158

REXX51 REXXTIP

162 Use MAKEBUF, DROPBUF Only When Needed
158, 158

REXX58 REXXTIP

162 VALUE is Faster Than GLOBALV For One Variable
158, 158

REXX22 REXXTIP

162 Minimize Wildcards
158, 158

REXX02 REXXTIP

163 Invoking Commands From XEDIT Macros
158, 158

REXX03 REXXTIP

163 To Invoke CP Command: ′CP cpcmd′
158, 158

REXX04 REXXTIP

163 To Invoke CMS Command: address ′COMMAND′ ′cmscmd′
158, 158

REXX05 REXXTIP

163 To Invoke an Exec: address ′COMMAND′ ′EXEC exec′
158, 158

REXX12 REXXTIP

164 To Invoke a Macro: ′MACRO macro′
158, 158

REXX13 REXXTIP

164 To Invoke an Editor Subcommand: ′COMMAND cmd′
158, 158

REXX14 REXXTIP

164 Additional Editor Short Cuts
158, 158

REXX56 REXXTIP

164 Use EXTRACT Instead of CMS Stack
158, 158

REXX16 REXXTIP

164 SET SYNONYM OFF
159, 159

REXX15 REXXTIP

165 Combine Commands, Macros Sent to the Editor
159, 159

REXX17 REXXTIP

165 Use Implicit SET for SET Editor Command
159, 159

REXX18 REXXTIP

165 Abbreviate XEDIT Subcommands and Parameters
159, 159, 164

REXX19 REXXTIP

165 Accept XEDIT Command Defaults
159, 159

REXX29 REXXTIP

165 Minimize Space
159, 159

REXX71 REXXTIP

165 Remove Blank Lines
159, 159

REXX72 REXXTIP

165 Minimize Spaces Before Code on Line
159, 159

REXX73 REXXTIP

166 Do Not Space Before the Continuation Character

				159, 159
REXX30	REXXTIP	166	Working With Comments	159, 159
REXX74	REXXTIP	166	Minimize Characters in Comments	159, 159
REXX78	REXXTIP	166	Execution Path Length = 0 for Code Comments	159, 159
REXX75	REXXTIP	166	Minimize the Number of Consecutive Line Comments	159, 159, 167, 167
REXX76	REXXTIP	167	Use Multiline Comments	159, 159
REXX77	REXXTIP	167	Locate Line Comments Before Labels	159, 159
REXX79	REXXTIP	167	Keep Line Comments Out of Hot Paths	159, 159
REXX39	REXXTIP	167	Working With IF and SELECT Statements	159, 159
REXX31	REXXTIP	167	ELSE Path is Faster for IF-THEN-ELSE	159, 159
REXX36	REXXTIP	167	Order of OR Expressions for THEN path Is Irrelevant	159, 159, 168
REXX38	REXXTIP	168	Use SELECT for OR-THEN Path	159, 159, 167
REXX35	REXXTIP	168	Order of AND Expressions for ELSE Path is Irrelevant	159, 159, 169
REXX81	REXXTIP	169	Use SELECT for AND-ELSE Path	159, 159, 168
REXX82	REXXTIP	169	Use Nested IFs In Place of SELECT	159, 159
REXX40	REXXTIP	170	Use IF Flag THEN ... ELSE	159, 159, 170
REXX65	REXXTIP	170	Use ′DO flag′ In Place of ′IF flag THEN′	159, 159, 170
REXX50	REXXTIP	171	SELECT: Put Most Likely WHENs Higher	159, 159
REXX43	REXXTIP	171	Working With Variables	159, 159
REXX83	REXXTIP	171	To Initialize All Compound Variables, aaaa. = 5	159, 159
REXX84	REXXTIP	171	Use PARSE to Initialize Variables	159, 159
REXX54	REXXTIP	171	Use Short Variable Names	159, 159
REXX62	REXXTIP	171	Start Parts of Compound Name Tail With a Digit	159, 159
REXX85	REXXTIP	172	Use PARSE VAR to Process Words in a Variable	159, 159
REXX86	REXXTIP	172	Other Tips	159, 159
REXX10	REXXTIP	172	Set Default REXX Environment: address ′COMMAND′	156, 159, 159
REXX42	REXXTIP	172	Built-in Functions Are Generally More Efficient.	159, 159
REXX41	REXXTIP	172	Remove Loops When Possible	

Footnotes

id	File	Page	References
M4VSEF1	CONFIG1	56	2 56, 81
NUVDFN1	NUVDISK	77	3 77
NUVDFN2	NUVDISK	79	4 79
M4VSEF2	CONFIG2	81	5
CMSTFN1	CMSTDSK	111	6 111

Revisions

id	File	Page	References
REV1	21SETUP	i	
REV2	21SETUP	i	
REV3	21SETUP	i	

Spots

id	File	Page	References
MMIGREF	MMIGSUM	68	(no text) 70, 72, 74
DER	MMIGSUM	72	(no text) 70, 71

Tables

id	File	Page	References
MIGSUM3	MMIGSUM3	69	16
MIGSUM1	MMIGSUM1	70	17 70, 73, 74, 75
MIGSUM2	MMIGSUM2	74	18 73
CMSFSI1	REXXL2	126	31
ITRTAB	PBTTABLE	131	32
PCAPTB	PCAPTABL	133	33 131
VARTAB1	VARTAB1	182	39 181
VARTAB2	VARTAB2	184	40 181
FS7FT1	21FS7F	192	42 192
FS7FT2	21FS7F	194	43

194
IOBT1 WORKOFV
204 44
204

Processing Options

Runtime values:

Document fileid	REPORT21 SCRIPT
Document type	USERDOC
Document style	DEFAULT
Profile	EDFPRF40
Service Level	0013
SCRIPT/VS Release	4.0.0
Date	93.06.24
Time	10:12:06
Device	38PPN
Number of Passes	3
Index	YES
SYSVAR D	SB
SYSVAR G	INLINE
SYSVAR S	OFFSET
SYSVAR X	YES

Formatting values used:

Annotation	NO
Cross reference listing	YES
Cross reference head prefix only	NO
Dialog	LABEL
Duplex	SB
DVCF conditions file	(none)
DVCF value 1	(none)
DVCF value 2	(none)
DVCF value 3	(none)
DVCF value 4	(none)
DVCF value 5	(none)
DVCF value 6	(none)
DVCF value 7	(none)
DVCF value 8	(none)
DVCF value 9	(none)
Explode	NO
Figure list on new page	YES
Figure/table number separation	YES
Folio-by-chapter	NO
Head 0 body text	(none)
Head 1 body text	(none)
Head 1 appendix text	Appendix
Hyphenation	YES
Justification	NO
Language	ENGL
Keyboard	395
Layout	OFF
Leader dots	YES
Master index	(none)
Partial TOC (maximum level)	4
Partial TOC (new page after)	INLINE
Print example id's	NO
Print cross reference page numbers	YES
Process value	(none)
Punctuation move characters	.,
Read cross-reference file	(none)
Running heading/footing rule	NONE
Show index entries	NO
Table of Contents (maximum level)	3
Table list on new page	NO
Title page (draft) alignment	CENTER
Write cross-reference file	(none)

Imbed Trace