NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PERFORMANCE ANALYSIS OF COMPUTER INSTALLATIONS
VIRTUAL MACHINE / 370 (VM/370)

by

Waldo Marmanillo Lazo

December 1981

Thesis Advisor: N.F. Schneidewind.

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Performance Analysis of Computer Installations Virtual Machine / 370 (VM/370)		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Waldo Marmanillo Lazo		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1981
		13. NUMBER OF PAGES 169
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Performance evaluation of computers

Virtual Machine / 370

Performance measurement and analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Highlights of the IBM 4341 and IBM 3033 AP systems are presented, with emphasis of performance aspects. An analysis of performance of the Virtual Machine Facility 370 (VM/370) is performed. The main efforts are (1) to present a methodology based on performance measurement and analysis techniques, trying to relate the trends in the data to the characteristics of the system, and thus gain an insight into what might cause the system to saturate and its performance to degrade, (2) analyze the statistical

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

correlations among performance and resource usage variables in order to estimate the degree of association among these variables, (3) identify those variables that are good indicators of system load, (4) formulate regression equations for forecasting the system performance.

This thesis is an effort toward the development of performance and resource usage forecasting equations, and a model for analyzing computer performance and resource allocation, of computer systems using VM/370.

Approved for Public Release, Distribution Unlimited

Performance Analysis of Computer Installations
Virtual Machine / 370 (VM/370)

by

Waldo Marmanillo Lazo
Lieutenant, Peruvian Navy
B.S., Peruvian Naval Academy, 1974

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN COMPUTER SCIENCE

AND

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
December 1981

ABSTRACT

Highlights of the IBM 4341 and IBM 3033 AP systems are presented with emphasis on Performance aspects. An analysis of Performance of the Virtual Machine Facility 370 (VM-370) is performed. The main efforts are (1) to present a methodology based on performance measurement and analysis techniques, trying to relate the trends in the data to the characteristics of the system, and thus gain an insight into what might cause the system to saturate and its performance to degrade, (2) analyze the statistical correlations among performance and resource usage variables in order to estimate the degree of association among these variables, (3) identify those variables that are good indicators of system load, (4) formulate regression equations for forecasting the system performance.

This thesis is an effort toward the development of performance and resource usage forecasting equations, and a model for analyzing computer performance and resource allocation, of computer systems using VM/370.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

The basic objective of computer center management is to provide high computer system performance at a reasonable cost under conditions of fluctuating workload and fixed computer resources [Ref. 1]. In order to satisfy this objective, it is necessary to forecast the performance and resource utilization which would result from a permanent and significant change in workload, if resources remain unchanged. If projected performance is unsatisfactory or the anticipated resource utilization is low, the forecast provides a warning that resources must be expanded or contracted, respectively. Once the condition of saturation or under utilization has been anticipated, it is necessary to forecast the performance and resource utilization which would be obtained when resources are changed.

For the Trident Submarine Command and Control System, the Tactical Software Support System (TS3) Resource Management Plan defines the specifications, general procedures and activities relating to the initiation and continuing operation of a resource management function.

This function's responsibilities will be to collect, retain, process, and analyze data relating to the utilization of TS3 laboratory resources and to communicate pertinent information to the TS3 and Trident Command and Control System (CCS) user community management.

The purpose of the Resource Management Plan defines directions and activities relating to the monitoring and reporting of computer system resource utilization and performance, addressing also the application of forecasting for capacity planning.

The TS3 laboratory uses IBM 4341 systems with the

Virtual Machine Facility 370 (VM/370) operating system, the one which manages the resources of the 4341 system complex in such a way that multiple users have a functional simulation of a computing system (a virtual machine) at their disposal.

This thesis is an effort toward the development of performance and resource usage forecasting equations and a model for analyzing computer performance and resource allocation for systems using VM/370.

The main efforts in the analysis of VM/370 system are: (1) to present a methodology based on performance measurement and analysis techniques, trying to relate the trends in the data to the working of the system, and thus gain an insight into what might cause the system to saturate and its performance to degrade, (2) analyze the statistical correlations among performance and resource usage variables in order to estimate the degree of association among these variables, (3) identify those variables that are good indicators of system load, and (4) formulate regression equations for forecasting the computer system performance.

Chapter II presents the highlights of the IBM 4341 System, with emphasis on some performance aspects. Chapter III presents the highlights and performance considerations of the Virtual Machine/System Product (VM/SP), which is planned to replace the VM/370 as an operating System, at the TS3 laboratory.

For the data collection the IBM 3033 AP system using VM/SP in the W.R. Church Computer Center of the NPS was used. Chapter IV presents some highlights of this system and its actual configuration. Chapter V covers the VM/SP performance measurement tools. Chapter VI presents the methodology used for the performance analysis and gives the

results obtained.

Appendices A to D contain the executive programs, program listings, sample of the data obtained at the data collection stage and a sample of one observation output.

II. IBM 4341 HIGHLIGHTS

A. IBM 4300 SERIES

1. Introduction

The IBM 4300 Series, which was announced in January 1979 initially consisted of two central processors, the 4331 and the 4341, together with five new peripheral devices and three enhanced operating systems. In May 1980, IBM filled the large performance gap between the two original processors by adding the 4331 Model Group 2. This processor has twice the processing power and up to four times the main memory capacity of the original 4331 , which is now designated the 4331 Model Group 1.

The most important aspects of the 4300 Series product line are [Ref. 2]: 1) the strikingly improved price/performance it offers; 2) the advanced technology employed to achieve those price/performance gains; and 3) the accompanying changes in IBM software pricing and support policies.

In terms of hardware performance per dollar, the 4300 Series processors offer approximately a four-fold increase over the corresponding System/370 processors, [Ref.3]. The software announcements that accompanied the 4300 Series introduction indicate a continuing IBM commitment to improve both the functionality of its software and the support it provides to users of these products. At the same time, the software policy is clearly designed to ensure that increased software and support cost will at least partially offset the savings in hardware costs that the new computers will bring to IBM users.

2. Processors

The 4300 Series central processors can operate

either in a System/370 - compatible mode or in an Extended Control Program (ECPS) mode. The latter mode takes full advantage of the extensive microcoding available in these machines to reduce operating system overhead and improve system throughput.

In comparison with the 370/138 (Jun. 1976), memory packaging on the 4300 Series is 32 times denser, and because of this, the 4331 Model Group 1 requires up to 70% less power, and the 4341 requires over 50% less power than the 370/138

All three of the 4300 Series processors share these common features:

- The system/370 Universal Instruction Set.
- Channels with virtual storage addressing.
- Maintenance support functions including a support processor and remote support facility.
- Store and fetch storage protection.
- Byte - oriented operands.
- Clock comparator, CPU timer, time of day clock, interval timer.
- PSW Key handling, control registers.
- Extended-precision floating point.
- Program event recording.
- and machine check handling,

According to IBM, the 4331 Model Group 1 is designed for the first-time computer user, such as a department or branch office within a larger enterprise, that could benefit from data base/data communications, interactive and distributed processing capabilities; it is also meant to replace many of the remaining IBM System/360 computers still in service. The 4331 Model Group 1 can operate as a stand-alone unit, or it can be linked to other 4300's or attached to a central System/370 host.

Peripheral and communications equipment can be connected to the 4331 Model Group 1 by means of one byte multiplexer channel, one block multiplexer channel, and several integral adapters. Data rates on the two channels may not exceed 500k bytes per second, which precludes the connection of high-speed disk or tape units. The optional DASD adapter, however, permits direct connection of up to four direct-access storage devices for a maximum on-line disk storage capacity of over 9 billion bytes.

The 4331 Model Group 2 processor appeared in May 1980, featuring major improvements in performance, memory capacity, and input/output capabilities over the 4331 Model Group 1. The rated instruction execution speed of the 4331 Model Group 2 is twice that of the 4331 Model Group 1 and a little over one-half that of the 4341. The new processor is offered in 4 models with memory capacities of 1,2,3 and 4 Megabytes.

The 4331 Model Group 2 can be equipped with the same integrated peripheral adapters as the Group 1 processor, plus an optional second DASD adapter and greatly improved I/O channel capabilities. One high-speed block multiplexer channel can handle a data transfer rate of up to 1.86 million bytes per second, permitting the attachment of high-speed disk storage units.

An installed 4331 Model Group 1 processor can be field-upgraded to a Group 2 processor in aproximately 13 to 16 hours.

The IBM 4341 processor is available in two models with main memory capacities of two and four megabytes. It features a lower purchase price and an instruction execution speed up to 3.2 times as fast as a System/370 Model 138 with 1 Mbyte of memory. The 4341's performance capability falls between that of the 370/148 and the 370/158-3. IBM describes the 4341 as particularly suitable for experienced

intermediate system users who need increased processing power, and those who could benefit from distributed applications that require more capacity.

None of the integrated peripheral adapters used on the 4331 processors is available for the 4341. Instead, all peripheral and communications devices are connected via standard I/O channels and control units.

Along with the 4300 Series processors, IBM introduced five new peripheral devices: the 3310 and 3370, Direct-Access Storage Devices, the 3880 Control Storage, and the 3205 Model 5 and 3262 printers.

3. Software and Support

Not since the System/360 unveiling in 1964 had IBM made an announcement that could impact users of its operating system software to such a magnitude as the software and support announcements that accompanied the January 1979 unveiling of the 4300 Series computer line. These announcements spelled out the operating environments of the future, and clearly identified the life spans of several existing operating systems. [Ref.4]

a. Operating Systems

There are three new system control program environments:

- An extended version of DOS/VS, called DOS/VS extended, or simply DOS/VSE.
- a new version of OS/VS1 labeled Release 7.
- and Release 6 of the Virtual Machine Facility/370 (VM/370).

These system control programs (SCP'S) support a series of new program products many geared specifically to support the 4300 Series computers.

The first SCP environment is DOS/VS-EXTENDED

(DOS/VSE) which is said to be a major expansion of DOS/VS, incorporating new functional and I/O support. DOS/VSE provides only limited multiprogramming capabilities unless the user acquires the DOS/VSE/advanced function product, an independently priced adjunct that allows the DOS/VSE user to employ up to 12 partitions and also makes it possible to incorporate many of the new program products available with the system.

This extended disk-resident operating system provides enhancements over IBM's older DOS/VS in the specific areas of processor support, hardware features, device support, usability inprovements, and serviceability. DOS/VSE supports the System/370 mode and the ECPS: mode of the 4300 processors. When operating in ECPS:VSE mode, DOS/VSE takes adavantage of the 4300 processor's concept of relocating channels and page management. To support the hardware extensions to page management, the DOS/VSE assembler has additional privileged instructions. The basic DOS/VSE system provides the capability for multiprogramming of five concurrent job streams, which will typically include the VSE/POWER spooler, a real-time subsystem such as CICS/VS, one or two batch job streams, and an unscheduled work partition for jobs that require fast turnaround. The system's capabilities can be significantly expanded through the addition of the VSE/Advanced Functions Program Product.

The second SCP environment is OS/VS1 Release 7, and IBM says that this support is of particular importance in a distributed data processing environment, since it will generally provide a high level of compatibility with an MVS host system. As with DOS/VSE and VM/370, OS/VS1 Release 7 can run in ECPS mode with the ECPS:VS1 feature on either the 4331 or 4341 processor or in 370 mode.

Two of the communications-oriented enhancements available with OS/VS1 Release 7 include: RES(REMOTE ENTRY

SERVICES), a component of OS/VS1 which allows jobs and commands to be submitted from remote terminals, with output returned, and HRNES (Host Remote Node Entry System), which allows and OS/VS1 system to be a remote job entry station to any MVS/JES2 or SVS/HASP System or to another OS/VS1 system. Operation is not dedicated; batch and on-line applications can be run concurrently.

In the third SCP environment, VM/370 Release 6, the 4300 user can operate in mixed-mode environments where CMS interactive computing is combined with a guest SCP (DOS/VSE or OS/VS1) on the 4300 processors.

In addition to supporting DL/1 DOS/VS and VSE/VSAM, VM/370 Release 6 supports VS/IFS (Interactive File Sharing), which allows multiple CMS users to share VSAM data sets. VM/Directory Maintenance, for management of the VM/370 Directory; Display Management of the VM/370 Directory, Display Management System/CMS; the query by example (QBE) interactive end-user query language; SPF/CMS (Structured Programming Facility/CMS); the DES (Display Editing System); high-level language support; and the System Installation Productivity Options/Extended (IPO/E).

b. Environment Types

The 4300 Series computers support four types of environment [Ref.5]: stand-alone, distributed applications, distributed data applications, and distributed networks.

In the stand-alone system environment, compatible growth is provided from the 4331 to the 4341 or 303x systems operating under DOS/VSE, VM/370 Rel 6, or OS/VS1 Rel 7. Growth through VSE/POWER shared spooling support or through VM/370 RSCS Networking.

In a distributed application environment, [Ref.6] host-connect applications may vary from periodic transmission of summary data between the 4300 and the host

system to a continuous connection offering RJE and/or passtrough capabilities. RJE is provided by DOS/VSE SNA and BSC program products plus VM/370 RSCS Networking and OS/VS1 HRNES IUP. Passtrough facilities are supported by ACF/VTAME, ACF/VTAM/MSNF, and VSE/3270 Bisync Pass-through.

Data that is most frequently used locally may be stored on the 4300's own direct access storage devices, with transaction-by-transaction access to the central host data base as needed in distributed data applications. Here CICS/VS Intersystem Communications with DL/1 and IMS Multiple Systems Coupling provides support.

In a distributed network, communication can be established between local or remote 4300's to the host computer, or to IBM 8100 Information Systems. Transactions from the 8100's to CICS/VS Intersystem Communications are supported by the 8100 DPPX Host Transaction Facility. DDPX also supports RJE to OS/VS1 RES and VM/370 RSCS Networking systems.

c. Compatibility

Any program written for an IBM System/370 computer will operate on a 4300 Series Processor in System/370 mode, provided that it is not time-dependent; does not depend on system facilities such as storage size, I/O equipment, optional features, etc., being present when the facilities are not included in the configuration; does not depend on system facilities such as interruptions, operation codes, etc., being absent when the facilities are included in the 4300 Processor; and does not depend on results or functions which IBM specifies to be unpredictable or model-dependent.

Any program written for a system/360 will operate on a 4300 Series processor in System/370 mode, provided that it follows the above rules and does not depend

on functions that differ between the system/360 and System/370.

Any program written for the IBM 4331 Processor in ECPS:VSE mode or System/370 mode will operate on the 4341 processor provided it follows the above rules.

B. THE 4341 PROCESSOR

1. Highlights

The 4341 Processor is an intermediate-scale, general purpose processor. It offers System/360-and System/370 compatible architecture, a new architecture that provides new functions, and a new level of price performance for intermediate system users made possible by the use of large-scale integrated technology. The 4341 Processor provides the range of commercial and scientific data processing capabilities offered by System/360 and System/370.

In addition to supporting virtual storage, a virtual machine environment is supported by Virtual Machine Facility/370 (VM/370), the successor to CP-67/CMS for System/370. VM/370 provides interactive computing via its conversational Monitor System (CMS) component and remote spooling via its Remote spooling Communications Subsystem (RSCS) component.

a. Instruction Processing Features

The following are instruction processing features of the 4341 Processor [Ref.7]:

- Implementation of a System/370 mode and an ECPS:VSE Mode, both of which support virtual storage, is standard. The major difference between the two modes is the way in which address translation is performed to support a virtual storage environment (ECPS:VSE mode is specifically designed to be utilize with the DOS/VSE operating system to provide increased processor performance when compared

to that achieved using DOS/VSE executing with System/370 mode in effect).

- The cycle time of the 4341 processor varies from 150 to 300 nanoseconds.
- The standard 4341 processor instruction set consists of the entire instruction set provided for System/370 (except for multiprocessing and direct control instructions) and several new control instructions that can be utilized only when ECPS:VSE mode is in effect.
- Precision of up to 28 hexadecimal digits, equal to up to 34 decimal digits, is provided by the extended precision data format.
- An interval timer of 3.3 milliseconds resolution, which can improve job accounting accuracy, is a functional feature of the 4341 processor.
- A time of day clock is included as a standard feature to provide more accurate time-of-day values than does the interval timer. This clock has a one-microsecond resolution.
- A CPU timer and clock comparator are standard. The CPU timer provides an interval timing capability similar to that of the interval timer, but has a much larger capacity than the latter and is updated every microsecond, as is the time-of-day clock. The clock comparator can be used to cause an interruption when the time-of-day clock passes a specified value. These terms provide higher resolution timing facilities than the interval timer and enable more efficient timing facility routines to be used.
- Program event recording is standard and is designed to be used as a problem determination aid. This feature includes hardware that monitors the following during program execution: succesful branches, the alteration of general registers, and instruction fetching from, and alterations of, specified areas of processor storage. It

only works in EC mode. In the 4341 Processor, additional processor time is required to execute instructions when program event recording is operative.

- ECPS:VS1 and ECPS:VM/370 features are standard, they increase the performance of a VS1 and VM/370 Operating system (respectively) when it operates in a 4341 Processor, since these functions cause certain control program routines to execute in hardware instead of as routines written using 4341 Processor instructions.
- A reduction of up to 7 percent of supervisor state processor busy time has been measured when ECPS:VS1 is utilized by the 4341 Processor as compared to the same version of OS/VS1 operating without the assist activated. A reduction of up to 84 percent of the supervisor state processor time used by CP has been measured when ECPS:VM/370 is utilized as compared to the same CP operating without the assist activated.

b. Storage Features

The following are significant storage features of the 4341 Processor:

- All storage in the 4341 processor-processor (main) control, high speed buffer, and local memory is implemented using monolithic technology instead of discrete ferrite cores. The technology used for processor storage in the 4341 processor provides a much denser storage chip (64K or 16k bits per chip) than is used in most system/370 processors (2k bits per chip).
- A two level storage system is implemented, consisting of large processor storage used as backing storage for a smaller high-speed buffer storage. The instruction processing function works mostly with the buffer so that the effective processor storage cycle is a fraction of the actual processor storage cycle. Eight thousand bytes of

high-speed buffer storage is a standard. Data is fetched from the buffer at a rate of 225 nanoseconds for a doubleword.

c. Channel Features

The following channel features are provided for the 4341 Processor:

- Two channel groups are available for the 4341 Processor. The standard channel group consists of one byte multiplexer and two block multiplexer channels. The optional channel group consists of three block multiplexer channels or one byte and two block multiplexer channels. The standard byte multiplexer channel has a 16-KB/sec maximum byte mode data rate for four-byte transfer operations. For burst mode operations, a maximum data rate of 1 MB/sec is possible for a buffered device.

  Optionally, one channel-to-channel adapter can be installed in a 4341 Processor and attached to any block multiplexer channel. The adapter can be used to connect the channel in the 4341 Processor to a channel in a System/360, a System/370, or another 4341 Processor.

d. Monitoring Feature

This is standard in the 4341 Processor and functionally identical to the System/370 monitoring feature. This feature provides the capability of monitoring the occurrence of programmed events. For example, monitoring can be used to perform measurement functions (how many times a routine was executed) or for tracing functions for the purpose of program debugging (which routines were executed).

The Monitor CALL instruction is provided with the monitoring feature. Execution of this instruction indicates the occurrence of one of the events being monitored. The operands of the Monitor CALL instruction

permit specification of up to 16 classes of events, each class with up to 16 million unique types of events. When a Monitor CALL instruction is executed, a program interruption occurs, if the monitor class indicated is specified, and the event identification (class and type) is stored in the lower fixed storage area.

Both the PER facility and the monitoring feature are provided for debugging purposes. The two features differ from one another in (1) the number of events that can be defined, (2) whether the events are defined by the hardware or the programmer, and (3) wheter the hardware or the programmer checks for the events and causes the interruptions. When PER is used, once the events to be monitored have been designated by the user, processor hardware checks for the occurrence of the events and causes the interruption. When the monitoring feature is used, the user defines the events to be monitored (up to 16 classes with up to 16 million codes each, instead of four events), and causes the program interruption by placing MONITOR CALL instructions at the desired places within the program.

e. I/O Features

The fast internal performance of the 4341 Processor, together with the expanded use of multiprogramming, requires that more data be available at a faster rate. The 4341 Processor supports more and faster concurrent high speed I/O Operations than similar models (370/50, 370/65). It also provides the block multiplexing capability, which is not available in those models. The I/O features of the 4341 processor provide:

- Attachment of the 3505 Reader and 3525 Punch with variety of models, that can operate at 800 and 1200 cards per minute, respectively.
- Attachment of variety of printers. The 3203 Model 5 with

print speed of 1200 alphanumeric lines per minute. The high-speed 3211 printer with print speed of 2000 alphanumeric lines per minute. The 3800 Printer subsystem, for very high-speed printing (up to 10,020 lines per minute or 20,040 lines, with double number of lines per inch).

- Low cost attachment of up to three 3278 Model 2A displays and/or 3287 Model 1 or 2 printers.
- Support of synchronous data link control communications for remote units attached via the 3704/3705 communications controllers.
- Attachment of high-speed, high capacity, direct access devices, such as 3370, 3330-series, 3340/3344, 3350, and the 2305 Model 2.
- Attachment of high-speed tape units, such as the 3420 models 4, 6 and 8.
- Potential increases in channel throughput via use of block multiplexing and rotational position sensing to improve effective data transfer rates.
- A significantly high attainable aggregate I/O data rate to balance the higher performance capabilities of the 4341 Processor. (Maximum 9MB/sec, with 5 block multiplexer channels).

2. Comments

Since hardware features and programming systems support for the 4341 Processor are upward compatible with those of System/360, the 4341 Processor offers Model 50 and 65 users significantly expanded computing capabilities without the necessity of a large conversion effort. Little or no time need be spent modifying operational System/360 or System/370 application programs or the IBM 1400 programs currently being emulated.

Existing processor-bound System/360 programs can

execute faster in a 4341 Processor because of the significantly increased internal performance of the 4341 Processor, while I/O-bound programs can benefit from the use of more processor storage, faster channel capability, block multiplexing, and faster I/O devices. The 4341 Processor also offers economical and flexible entry into communications-based applications.

The increased power and new functions of the 4341 Processor provide the base for expanded application installation and penetration of previously marginal application areas. New application installation and transition to online operations can be easier when a virtual storage environment is implemented. The greatly improved price performance of the 4341 Processor offers the System/360 and System/370 user the opportunity to widen his data processing base for a significantly lower cost than was previously possible.

For large installations that want undisrupted growth and decentralization of their data processing facilities, the 4341 Processor provides economical and easy entry into (or expansion of) distributed data processing operations.

A virtual storage environment is designed primarily to provide new functional capabilities for the installation as a whole, although performance gains are possible for installations with particular environmental characteristics.

The general functional aims of IBM-supplied virtual storage operating systems are (1) to use new hardware features and additional control program processing to support certain facilities that are not possible in a nonvirtual storage environment because of real storage restraints and (2) to handle other functions that must be performed by installation personnel (programmers, operators, and system designers) when virtual storage and address translation are not used.

It is also important to note that, while a virtual storage operating system permits an installation to be independent of real storage restraints to a large degree and enables real storage to be utilized more efficiently, the performance of the system and specific advantage that can be achieved still depend largely on the amount of real storage present in the system and on the computing speed of the processor, among other things. Hence, virtual storage and an address translation capability are not a substitute for real storage. Rather, they provide an installation with greater flexibility in the tradeoff between real storage size and function or performance.

The degree to which a particular installation experiences the potential benefits of a virtual storage/address translation environment is highly system configuration dependent and application dependent (number, type, complexity of applications installed or to be installed). In addition, consideration must be given to the system resources that are specifically required to support a virtual storage environment.

III. VIRTUAL MACHINE/SYSTEM PRODUCT (VM/SP) HIGHLIGHTS AND PERFORMANCE CONSIDERATIONS

A. HIGHLIGHTS

Virtual Machine/System Product (VM/SP) in conjunction with VM/370 Release 6 is a System Control program. It manages the sources of an IBM System/370, or 4300, or 303x system complex in such a way that multiple users have a functional simulation of a computing system (a virtual machine) at their Disposal.

That is, the virtual machine runs as if it were a real machine simulating both hardware and software resources of the system. These simulated resources can be shared either with other virtual machines or alternately allocated to each machine for a specified time. Furthermore, virtual machines can run the same or different operating systems simultaneously. Thus, the individual user can create and adapt his virtual machine to meet his own special needs.

VM/370 Release 6 System consists of four components: the Control Program (CP), Conversational Monitor System (CMS), Remote Spooling Communications Subsystem (RSCS), and Interactive Problem Control System (IPCS). Each of these components control its unique part of the system. Together, these components provide the virtual machine with time sharing, remote spooling, and problem reporting for System/370 uniprocessor, attached processor, and multiprocessor systems. Two of these components, CP and CMS, have been extensively modified with new functions and efficiencies and integrated into a VM/370 Release 6 base. In publications, this collective package, that is, CP CMS, RSCS and IPCS, is simply referred to as VM/SP.

Control Program

The control program (CP) executes in a real machine controlling the resources of that machine. CP is the vehicle that is used to create concurrent virtual machines.

Conversational Monitor System

CMS is a single-user operating system designed to operate in a virtual machine. CMS provides a wide range of general-purpose, conversational time sharing functions.

Remote Spooling Communications Subsystem

RSCS, VM/370 Release 6 component, is a single-user operating system that runs under CP. RSCS executes in one or more virtual machines and transfers data between virtual machines and remote users.

Interactive Problem Control System

IPCS, a VM/370 Release 6 component, is a group of commands and controls that execute under CMS to provide problem analysis and management facilities. IPCS standardizes the process of reporting problems and includes a method for identifying duplicate problems within the system. It also provides the user with the capability of viewing and diagnosing CP abend dumps through the virtual machines operator's console.

1. The Virtual Environment

a. The Virtual Machine

A virtual machine is functionally equivalent to a real system. It has simulated hardware and software resources that operate in a real computer under CP. Each virtual machine is defined in the VM/SP directory; the directory describes its simulated storage, I/O devices and

console.

b. Virtual Storage

A virtual storage system can simulate real storage within a range of from 8 kbytes to 16 Megabytes (the maximum virtual storage size)

Virtual storage extends beyond the size of real storage and is not limited by the amount of real storage. It is highly probable for the combined virtual storage of several virtual machines to be greater than the real storage. Virtual Storage can be managed and protected through segmentation. Each segment is 64k, and there are from 1 to 256 segments depending on the size of virtual storage. As a storage protection feature for all virtual machines, page and segment tables are accessible only to CP. Generally, one virtual machine cannot access or alter the virtual storage of another virtual machine; however, mutually consenting users may share real-only virtual storage and real-write virtual storage.

c. Virtual Processor

CP provides each virtual machine with a single virtual processor to execute instructions and receive interruptions. In actuality, this virtual processor is the shared use of the real processor. CP simulates the privileged instructions, and the real processor executes the non privileged instructions. The virtual processor provided in the virtual machine is a uniprocessor simulation. Attached processor and multiprocessor simulation is not supported in the virtual machine environment, regardless of the installation's computer complex on which VM/SP is loaded.

d. Virtual System console

The virtual machine system console has three major communication functions. First, to communicate to CP, so it can provide to the virtual machine simulations of functions that are performed on real system consoles. Second, to provide the virtual machine user a means of dynamically altering specific attributes of a virtual machine. Third, to provide a means of communicating with the application program that is running in the virtual machine. To accomplish these functions requires a real terminal device. A virtual machine is "disconnected" by detaching its assigned supporting terminal console. It is done by CP command. Note, disconnecting the console does not negate current virtual machine processing.

e. Virtual I/O Devices

The virtual machine supports the same devices as a real machine: it is the virtual machine, not CP, that controls them. The I/O configuration must be defined by the user in the CP'S user directory entries. However, additional I/O requirements can be met dynamically by the user via CP commands.

The user also has the option of assigning different addresses to his virtual devices, or using those of the real devices. In either case, CP comments the virtual address to its real counterpart and performs any necessary data translation.

2. Virtual Machine Operating Systems

While the control program of VM/SP manages the concurrent execution of the virtual machines, it is also necessary to have an operating System manage the work flow within each virtual machine. Because each virtual machine executed independently of other virtual machines, each one can use either a different Operating system or different

releases of the same operating system.

A list of some of the operating systems [Ref. 8] that can execute in virtual machines, follows:

BATCH OR SINGLE-USER	INTERACTIVE
DOS	OS/PCP
DOS/VS	OS/MFT
DOS/VSE	OS/MVT
	OS/VS1
RSCS	OS/VS2 SVS
	OS/VS2 MVS
	OS-ASP

MULTIPLE - ACCESS

VM/370

TIME SHARING OPTION OF

OS DOS/VSE With VSE/ICCF

CONVERSATIONAL

CMS

With the exception of OS/PCP and CMS, these are all multiprogramming systems. However, when operating in a virtual machine, the user has the choice of running multiple partitions in one virtual machine (Similar to stand-alone operation) or single partitions in multiple virtual machines. When running multiple partitions in one virtual machine, multiprogramming and unit record spooling is done by both the Operating System and VM/SP. When running single partitions in multiple virtual machines, the need for multiple virtual storages places a burden on auxiliary storage. However, this can be alleviated by using shared systems.

a. Single - user Systems

Systems that can execute interactively by a

single user include the CMS and any operating system that can execute in a virtual machine. A time-sharing environment is created when VM/SP creates multiple virtual machines, each controlled by the same operating system. Thse systems operate concurrently with each other as well as with other conversational or batch systems.

b. Multiple-access Systems

Multiple-Access systems, such as MVS TSO, execute in one virtual machine and directly service many interactive terminals. To connect a terminal with the virtual machine, the user of a multiple-access system issues the DIAL command instead of the LOGON command.

Once his terminal is connected, the user issues only the commands associated with the multiple-access system.

3. VM/SP Applications

Using VM/SP, an installation can perform its work more efficiently and easily. Virtual machine applications aid in programming, operations, and interactive use.

a. System Programming

- Reducing the amount of hands-on testing time on the real machine.
- Testing new or modified SVC routines in a virtual machine.
- Generating and testing in a virtual machine either new independent component releases (ICRS) or new releases of an operating System.
- Debugging from a terminal device.

b. Application Programming

- Using the system product editor to create source programs and data files.

- Debugging from a terminal while under operating system control
- Providing faster turnaround time, more test periods per day, and a shorter development cycle.
- Designing application programs without real storage limitations.
- Defining minidisks and other virtual devices to design and test a slightly different or larger machine configuration before installing the hardware.
- Using SCRIPT/VS, a program product, for text preparation, to create an update program specifications.

c. Operations

An interactive virtual machine environment relieves problems of scheduling, support, and backup and expedites productions; some of these operational advantages are:

- Training operators in a virtual machine that is isolated from production virtual machines.
- Defining a virtual machine and its devices as backup to another real machine.
- Running different types of work concurrently on a single real machine.
- Executing many types of batch applications with no change to the program either in a individual virtual machine or in a virtual machine dedicated to executing programs in batch mode.

d. Backup System

An installation using VM/SP has more flexibility in using another System/370 computing system for backup. Neither the same System/370 model nor the same amount of real storage have to be part of the backup system. The backup system must include, but is not limited to the same

type and number of real devices as these virtual machines require. Also, the backup system must have a sufficient number of direct access storage drives so that the user volumes can be mounted.

E. GENERAL INFORMATION

The performance characteristics of an operating system, when it is run in a virtual machine environment, are difficult to predict. This unpredictability is a result of several factors: [Ref. 9]

- The System/370 model used.
- The total number of virtual machines executing.
- The type of work being done by each virtual machine.
- The speed, capacity, and number of the paging devices.
- The amount of fixed head paging storage.
- The amount of real storage available.
- The degree of channel and control unit contention, as well as arm contention, affecting the paging device.
- The type and number of VM/SP performance options in use by one or more virtual machines.
- The degree of MSS 3330 volume use.
- The order in which devices are selected for preferred paging and spooling.

The performance of a specific virtual machine may never equal that of the same operating system running standalone on the same System/370, but the total throughput obtained in the virtual machine environment may equal or better that obtained on a real machine.

When executing in a virtual machine, any function that cannot be performed wholly by the hardware causes some degree of degradation in the virtual machine's performance. As the control program for the real machine, CP initially processes all real interrupts. A virtual machine operating

system's instructions are always executed in problem state. Any privileged instruction issued by the virtual machine causes a real privileged instruction exception interruption. The amount of work to be done by CP to analyze and handle a virtual machine-initiated interrupt depends upon the type and complexity of the interrupt.

1. Program states

When instructions in the Control Program are being executed, the real computer is in the supervisor state; at all other times, when running virtual machines, the real computer is in the problem state. Therefore, privileged instructions cannot be executed by the virtual machine. Programs running on a virtual machine can issue privileged instructions; but such an instruction either (1) causes an interruption that is handled by the Control Program, or (2) is intercepted and handled by the processor, if the virtual machine assist feature or VM/370 Extended Control Program Support is enabled and supports that instruction. CP examines the operating status of the virtual machine PSW. If the virtual machine is in problem mode, the privileged interrupt is reflected to the virtual machine.

Only the Control Program may operate in the supervisor state on the real machine. All programs other than CP operate in the problem state on the real machine. All user interrupts, including those caused by attempted privileged operations, are handled by either the control program or the processor (is the virtual machine assist feature or VM/370 Extended Control-Program Support is available). Only those interrupts that the user program would expect from a real machine are reflected to it. A user program executes on the virtual machine in a manner identical to its execution on a real System/370 processor, as long as the user program does not violate the CP

restrictions.

2. Using Processor resources

CP allocates the processor resource to virtual machines according to their operating characteristics, priority, and the system resources available.

Virtual machines are dynamically categorized at the end of each time slice as interactive or noninteractive, depending upon the frequency of operations to or from either the virtual system console or a terminal controlled by the virtual machine.

Virtual machines are dispatched from one of two Queues, called Queue 1 and Queue 2. In order to be dispatched from either queue, a virtual machine must be considered executable (that is, not waiting for some activity or for some other system resource). Virtual machines are not considered dispatchable if the virtual machine [Ref.10].

- Enters a virtual wait state after an I/O operation has begun.
- Is waiting for a page frame of real storage.
- Is waiting for an I/O operation to be translated by CP and started.
- Is waiting for CP to simulate its privileged instructions.
- Is waiting for a CP console function to be performed.

a. Queue 1

Virtual machines in Queue 1 (Q1) are considered conversational or interactive users, and enter this Queue when an interrupt from a terminal is reflected to the virtual machine. The Q1 virtual machines are ordered by their deadline priorities in the dispatch list. A deadline priority is a value calculated by the fair share scheduler every time a user is dropped from a queue (queue drop time).

This value is based on paging activity, processor usage, the load on the system, and user priority. Deadline priority is used to determine when the user receives his next time slice.

A particular virtual machine's deadline priority for Q1 will be better (earlier) than its corresponding priority for Q2. The deadline priorities for all Q1 virtual machines are not necessarily better than the deadline priorities for all Q2 virtual machines.

Virtual machines are dropped from Q1 when they complete their time slice of processor usage, and are placed in an "eligible list". Virtual machines entering CP command mode are also dropped form Q1.

b. Queue 2

Virtual machines are selected to enter Q2 from a list of eligible virtual machines (the eligible list). The ordering of virtual machines on the eligible list and the dispatch list is determined on the basis of each virtual machine's deadline priority.

There are two lists of virtual machines in Q2; those in the eligible list and those in the dispatch list. Both lists are sorted by deadline priority. A particular deadline priority depends on many factors:

- The time-of-day the virtual machine last dropped from the dispatch list.
- The virtual machine's user priority
- The current load and number of virtual machines on the system
- The current resource utilization of the virtual machine

A virtual machine enters Q2 only if its working set size is not greater than the number of real page frames available for allocation at the time. The working set of a

virtual machine is calculated and saved each time a user is dropped from Q2. The working set size is a function of the number of virtual pages referred to by the virtual machine during its stay in Q2, and the number of its virtual pages that are resident in real storage at the time it is dropped from the queue.

If the calculated working set of the highest priority virtual machine in the eligible list is greater than the number of page frames available for allocation, CP continues to search the eligible list, in deadline priority order, for a virtual machine whose working set does not exceed the number of available page frames.

When a virtual machine completes its time slice of processor usage, it is dropped from Q2 and placed in the eligible list according to its deadline priority. When a virtual machine in Q2 enters CP command mode, it is removed from Q2.

To leave CP mode and return his virtual machine to the eligible list for Q2, a user can issue a CP command that thransfers control to the virtual machine operating system for execution (for example, beging, ipl, external, and RESTART).

Virtual machines in Q2 are considered to be noninteractive. In CP, interactive virtual machines (those in Q1), if any, are normally considered for dispatching before noninteractive virtual machines (Q2). This means that CMS users entering commands that do not involve disk or tape I/O operations should get fast responses from the VM/SP system even with a large number of active virtual machines. All virtual machines (Q1 and Q2) on the dispatch list are ordered by their deadline priority. There can be many instances where some virtual machines in Q2 are considered for dispatching before virtual machines in Q1 because of their user priority, current resource utilization level, or

for other reasons.

c. Deadline priority

The deadline priority is calculated at queue drop time by taking the current time-of-day (TOD) and adding a user bias factor, which is the product of the user bias ratio and the Q2 delay factor [Ref.11]. The Q2 delay factor, which is calculated dynamically based on configuration and load, is the average elapsed time required by a virtual machine to receive an amount of processor time equal to one Q2 time slice. In the scheduling algorithm calculations the Q2 delay factor is a scaling value used to adjust the calculation for configuration and load. Before adding it to the current time-of-day it is adjusted by the user bias ratio. The user bias ratio is less than 1, equal to 1, or greater than 1, depending on whether the particular virtual machine is currently receiving less than, equal to, or more than its specified amount of resources.

For Q1 virtual machines, the scaled bias is divided by 8 (since the Q1 processor usage time slice is 1/8th the Q2 time slice). The difference between scheduling a virtual machine in Q1 instead of Q2 is that it receives 1/8th the amount of processor, 8 times as often. Operating constantly in either queue, a virtual machine should receive the same amount of processor resources over an extended period of time. The only preference given Q1 virtual machines is when they are being moved from the eligible list to the dispatch list. They will be moved ahead of Q2 virtual machines with the same or even slightly better deadline priorities.

d. Queue 3

Q3 is an extension of Q2 sheduling. It helps to distinguish between non-interactive virtual machines and

those that are frequently switching back and forth between Q2 and Q1. Virtual machines that have cycled through at least eight consecutive Q2 processor time slices without a Q1 interaction are labeled Q3. Q3 virtual machines are kept in the same lists (or queues) as Q2 virtual machines and for most purposes are treated identicallly. The differences between Q2 and Q3 virtual machines are reflected in their deadline priority calculations and the amounts of such processor time they are allowed in queue. Q3 virtual machines are allowed eight consecutive Q2 processor time slices before they are dropped from queue. Because of the eight-fold increase in processor time allowed for each time in queue, the scaled bias is multiplied by eight before adding to the current time-of-day to form the deadline priority. Q3 virtual machines should receive eight times as much processor time each time in queue as Q2 virtual machines, but only 1/8th as often.

To reiterate the Q1/Q2 statement: operating constantly in any queue, a virtual machine should receive the same amount of processor resources over an extended period of elapsed time. This does not necessarily mean that a virtual machine will perform the same when operating in Q3 mode as when operating in standard Q2 mode. An amount of overhead (roughly proportional to the small number of resident pages) is used for each virtual machine when it drops from queue. When operating in Q3 mode, a virtual machine may perform much better than in normal Q2 mode because it is undergoing fewer queue drops. For some very large virtual storage programs, the total processor resources used has been cut in half by operating in Q3 mode as compared to standard Q2 mode.

C. VIRTUAL MACHINE I/O

To support I/O processing in a virtual machine, CP must

translate all virtual machine channel command word (CCW) sequences to refer to real storage and real devices and, in the case of minidisk, real cylinders. CP's handling of SIOS for virtual machines can be one of the most significant causes of reduced performance in virtual machines.

The number of SIO operations required by a virtual machine can be significantly reduced using: large blocking factors (up to 4076 bytes), preallocated data sets, virtual machine operating system options (such as chained scheduling in OS), or finally with the substitution of a faster resource (virtual storage) for I/O operations, by building small temporary data sets in virtual storage rather than using an I/O device.

Frequently, there can be a performance gain when CP paging is substituted for virtual machine I/O operations. The performance of an operating system such as OS can be improved by specifying as resident as many frequently used OS functions as are possible. In this way, paging I/O is substituted for virtual machine-initiated I/O. In this case the only work to be done by CP is to place into real storage the page that contains the desired routine or data.

Three CP performance options are available to reduce the CP overhead associated with virtual machine I/O instructions or other privileged instructions used by the virtual machine's I/O Supervisor:[Ref. 12]

1. The 'real' option removes the need for CP to perform storage reference translation and paging before each I/O operation for a specific virtual machine.
2. The virtual machine assist feature reduces the real supervisor state time used by VM/SP.
3. VM/370 Extended Control-Program Support (ECPS) further reduces the real supervisor state time used by VM/SP.

D. PAGING CONSIDERATIONS

When virtual machines refer to virtual storage addresses that are not currently in real storage, they cause a paging exception and the associated CP paging activity.

The addressing characteristics of programs executing in virtual storage have a significant effect on the number of page exceptions experienced by the virtual machine. When an available page of virtual storage contains only reentrant code, paging activity can be reduced, since the page, although referred to, is never changed, and thus does not cause a write operation to the paging device.

Virtual machines that reduce their paging activity by controlling their use of addressable space improve resource management for that virtual machine, the VM/SP system, and all other virtual machines. The total paging load that must be handled by CP is reduced, and more time is available for productive virtual machine use.

Additional dynamic paging storage may be gained by controlling free storage allocation, the amount of free storage allocated at VM/SP initialization time can be controlled by the installation.

CP provides three performance options, locked pages, reserved page frames, and a virtual=real area, to reduce the paging requirements of virtual machines. Generally, these facilities require some dedication of real storage to the chosen virtual machine and, therefore, improve its performance at the expense of other virtual machines.

1. Locked Pages Option

The LOCK command, which is available to the system operator (with privilege class A), can be used to permanently fix or lock specific pages of virtual storage into real storage. In so doing, all paging I/O for these page frames is eliminated. Only frequently used pages should be locked into real storage. Since page zero (first

4096 bytes) of a virtual machine storage is referred to and changed frequently, it should be the first page of a particular virtual machine that an installation considers locking. The virtual machine interrupt handler pages might also be considered good candidates for locking.

Other pages to be locked depend upon the work being done by the particular virtual machine and its usage of virtual storage.

Once a page is locked, it remains locked until either the user logs off or the system operator issues the UNLOCK command for that page.

2. Reserved Page Frames Option

A more flexible approach than locked pages is the reserved page frames option. This option provides a specified virtual machine with an essentially private set of real page frames, the number of frames being designated by the system operator when he issues the CP SET RESERVE command line. Pages will not be locked into these frames but they can be paged out only for other active pages of the same virtual machine.

This option is usually more efficient than locked pages in that the pages that remain in real storage are those pages with the greatest amount of activity at that moment, as determined automatically by the system. Although multiple virtual machines may use the lock option, only one virtual machine at a time may have the reserved page frames option active.

The reserved page frames option provides performance that is generally consistent from run to run with regard to paging activity.

3. Virtual=Real Option

This option eliminates CP paging for the selected

virtual machine. All pages of virtual machine storage, except page zero, are locked in the real storage locations they would use on a real computer. CP controls real page zero, but the remainder of the CP nucleus is relocated and placed beyond the virtual=real machine in real storage.

Since the entire address space required by virtual machine is locked, these page frames are not available for use by other virtual machines except when the 'real' machine is not logged on. This option increases the paging activity for other virtual machine users, and in some cases for VM/SP.

The 'real' option is desirable when running a virtual machine operating system (like DOS/VS or OS/VS) that performs paging of its own because the possibility of double paging is eliminated.

E. VM/SP PERFORMANCE CONSIDERATIONS

VM/SP provides a number of options [Ref.13] an installation may use to improve the performance of virtual machines and VM/SP. Several options improve the performance of installation specified virtual machines; other options improve the performance of all virtual machines and VM/SP. The options are:

- Favored execution
- User priority
- Reserved page frames
- 'real'
- Affinity
- Queue drop elimination
- Virtual machine assist
- Extended Control-Program Support.

Specifying a performance option may mean making a performance trade-off; improving the performance of one

virtual machine at the expense of VM/SP and other virtual machines.

1. <u>Favored execution</u>

The favored execution options allow an installation to modify the normal CP deadline priority calculations in the fair share scheduler to force the system to devote more of its processor resources to a given virtual machine than would ordinarily be the case: There are 2 options:

- The basic favored execution option
- The favored execution percentage option.

The basic, means that the virtual machine so designated is to remain in the dispatch list at all times, unless it becomes nonexecutable. When the virtual machine is executable, it is to be placed in the dispatchable list at its normal priority position. Multiple virtual machines can have the basic favored execution option set. However, if their combined main storage requirements exceed the system's capacity, performance can suffer because of trashing.

If the favored task is highly compute bound and must complete for the processor with many other tasks of the same type, the installation should define the processor allocation to be made. In this case, the favored execution percentage option can be selected. This option specifies that the selected virtual machine, in addition to remaining in Queue, is guaranteed a specified minimun percentage (from 1 to 100 percent) of the total processor time if it can use it. To select the favored execution option, specify the FAVORED operand on the SET command.

2. <u>User Priority</u>

The VM/SP operator can assign specific priority

values to different virtual machines. In so doing, the virtual machine with a higher priority is allocated a larger share of the system resources before a virtual machine with a lower priority. User priorities are set by the following class A command:

Set PRIORITY userid nn

Where userid is the user's identification and nn is an integer value from 1 to 99. The value of nn affects the user's dispatching priority in relation to other users in the system. The priority value (nn) is one of the factors considered in the calculation of the deadline priority. The deadline priority is the basis on which all virtual machines in the system are ordered on both the eligible list and the dispatch list. The deadline priority calculation is based on the assumption that the average or normal (default) user priority is 64.

3. Reserved Page Frames

VM/SP uses chained list of available and pageable pages. Pages for users are assigned from the available list, which is replenished from the pageable list.

Pages that are temporarily locked in real storage are not available or pageable. The reserved page function gives a particular virtual machine an essentially "private" set of pages. The pages are not locked; they can be swapped, but only for the specified virtual machine. Paging proceeds using demand paging with a "reference bit" algorithm to select the best page for swapping. The number of reserved page frames for the virtual machine is specified as a maximum.

4. Virtual=Real

For this option, the VM/SP nucleus must be reorganized to provide an area in real storage large enough

to contain the entire 'real' machine. In the virtual machine, each page from 1 to the end is in its true real storage location, only its page zero is relocated. The virtual machine is still run in dynamic address translation mode, but since the virtual page address is the same as the real page address, no CCW translation is required.

There are several considerations for the 'real' option that affect overall system operation [Ref.14].

5. Affinity

This option allows virtual machines that operate on attached processor or multiprocessor systems to select the processor of their choice for program execution. In application, the affinity setting of a virtual machine implies a preference of operation to either (or neither) processor. Affinity of operation for a virtual machine means that the program of that virtual machine will be executed on the selected or named processor. It does not imply that supervisory functions and the CP housekeeping functions associated with the virtual machine will be handled by the same processor.

In attached processor systems all real I/O operations and associated interrupts are handled by the main processor. Virtual I/O initiated on the attached processor that is mapped to real devices must transfer control to the main processor for real I/O execution. Therefore, benefits may be realized in a virtual machine "mix" by relegation those virtual machines that have a high I/O-to-compute ratio to the main processor, and those virtual machines that have a high compute-to-I/O ratio to the attached processor. Such decisions should be carefully weighed as every virtual machine is in contention with other virtual machines for resources of the system.

An important use of the affinity setting would be in

applications where there are virtual machine program requirements for special hardware features that are available on one processor and not the other.

6. Queue Drop Elimination

VM/SP attempts to optimize system throughput by monitoring the execution status of virtual machines. When a virtual machine becomes idle; VM/SP will drop it from the active queue. The virtual machines page and sequent tables are scanned, and resident pages are invalidated and put on the flush list.

VM/SP determines that a virtual machine is idle when it voluntarily suspends execution, and no high-speed I/O operation is active. Normally, this is an adequate procedure. However, in certain special cases, a virtual machine is determined to be idle and is dropped from the queue, but it becomes active again sooner than expected. If this cycle of queue dropping and reactivation is repeteadly executed, the overhead involved in invalidating and revalidating the virtual machine's page may become large.

The CP class A command "SET DROP userid ON/OFF allows the installation to control this situation. If SET QDROP OFF is in effect for a virtual machine, that virtual machine's pages are not scanned or flushed when the machine becomes idle. The page stealing mechanism is the only way the pages can be removed from storage. (Page stealing is invoked only if the flush list is empty).

7. Virtual Machine assist feature

The virtual machine assist feature is a processor hardware feature that improves the performance of VM/SP. Virtual storage operating systems, which run in problem state under the control of VM/SP, use many privileged instructions and SVCs that cause interrupts that VM/SP must

handle. When the virtual machine assist feature is used, many of these interrupts are intercepted and handled by the processor. Consequently, VM/SP performance is improved.

The Virtual Machine Assist Feature intercepts and handles interruptions caused by SVC's, invalid page conditions, and several privileged instructions.

Although the assist feature was designed to improve the performance of VM/SP, virtual machines may see a performance improvement because more resources are available for virtual machine users.

Whenever you IPL VM/SP on a processor with the virtual machine assist feature, the feature is available for all VM/SP virtual machines. However, the system operator's SET command can make the feature unavailable to VM/SP and, subsequently available again for all users.

The virtual machine assist features is not available to a second-level virtual machine, that is, a virtual machine that is running in a virtual machine.

8. VM/370 Extended Control-Program Support(ECPS)

ECPS, extended, for specific privileged instructions, the hardware assistance that the virtual machine assist features, provides. ECPS also provides hardware assistance for frequently used VM/SP functions. The use of ECPS improves VM/SP performance beyond the performance gains that the virtual machine assist feature provides.

ECPS consists of three functions:

- CP assist
- Expanded Virtual Machine assist
- Virtual interval timer assist.

CP assist provides hardware assistance for frequently used paths of specific CP functions.

Expanded virtual machine assist extends the hardware assistance that the VMAF provides for certain instructions and other privileged instructions.

Virtual Interval timer assist provides hardware updating of the virtual interval timer. Timer updating occurs only while the virtual machine is in control of the real processor. Virtual Interval timer assist updater the virtual timer at the same frequency hardware updates the real timer, 300 times per second. Thus, virtual interval timer assist updates the virtual timer more frequently than CP updates it. Because the timer is updated more frequently, accounting routines may be able to provide accounting data that is more accurate.

ECPS is controlled at two levels: The VM/SP system and the virtual machine.

At the VM/SP system level, ECPS is automatically enabled when the system is loaded. At the virtual machine level, whenever ECPS is enabled on the system, both expanded virtual machine assist and virtual interval timer assist are automatically enabled when you log on: There are different class G commands that allow the user to enable or disable the different ECPS assists.

F. OTHER SYSTEM FEATURES

VM/SP contains several other features [Ref.13] that expand the capabilities of operating systems running in virtual machines. They are:

- Virtual Machine Accounting
- Saved Systems
- Shared Systems
- Discontiguous saved segments
- Shared segment protection
- Virtual Machine communication facility.
- Inter-User communication vehicle.

1. *Virtual Machine Accounting*

VM/SP keeps track of a virtual machine's usage of system facilities and records accounting information whenever the use of some changeable resource is terminated.

2. *Saved System*

When initially loading an operating system into a virtual machine by device address, VM/SP reads the resident nucleus into real storage and writes it back out to the System paging device. Simultaneously it updates the virtual machine's paging tables.

In addition, at system generation, the system programmer can specify that the virtual machine contents of specified users be saved automatically on DASD if either VM/SP terminates the virtual machine or if VM/SP itself is terminated.

3. *Shared Systems*

A saved system can also share reentrant portions of its virtual storage among many concurrently operating virtual machines.

The greater the number of virtual machines that are using a shared system, the greater the storage savings, and the greater the probability that the shared pages will be frequently referenced. Frequently referenced pages tend to remain in real storage, thereby reducing paging activity. Less paging activity increases the efficiency of the processor.

4. *Virtual Machine Communication Facility*

VMCF, allows one virtual machine to communicate and exchange data with other virtual machines operating under the same VM/SP System.

5. Inter-User Communication Vehicle

The IUCV defines a precise protocol for communication between virtual machines operating under the same VM/SP system. In addition, it is possible for authorized virtual machines to communicate with console communication services portion of VM/SPE control program.

IV. THE 3033 ATTACHED PROCESSOR COMPLEX

A. THE 3033 PROCESSOR COMPLEX

1. General Definition of Multiprocessing

A multiprocessing configuration is one in which two or more processors are interconnected and execute two or more tasks simultaneously, one in each processor. Multiprocessing is a logical extension of multiprogramming in which two or more tasks operate concurrently in a single processor. In a multiprogramming environment one task executes at a time and only I/O operations for two or more tasks can operate simultaneously. In a multiprocessing environment both I/O operations and instruction execution for two or more tasks in the same or different programs can occur simultaneously, with each task executing in a different processor.

The hardware connection of the processors in a multiprocessing configuration is the means by which the processors communicate with each other in order to coordinate the activity of the multiprocessing configuration. A multiprocessing configuration can be tightly or loosely coupled or can include a combination of both loosely and tightly coupled processors.

A tightly coupled multiprocessing configuration is one in which (1) the processors share access to all the processor storage available in each system, (2) processor-to-processor communication is accomplished via the storing of data in shared storage and via direct processor-to-processor signals (both program- and hardware-initiated), and (3) a single control program is used. The 3033 Multiprocessor Complex is, therefore, a tightly coupled multiprocessing configuration, as is the 3033 Attached

Processor Complex.

A loosely coupled multiprocessing configuration is one in which (1) processors are coupled via channel-to-channel connections, (2) each processor has its own control program, and (3) a single system scheduling and operational interface is optional.

The objective of coupling multiple systems to form a multiprocessing configuration is to obtain a configuration that combines advantages of a single processor environment with those of an uncoupled multiple processor environment.

A single processor environment offers the following advantages:

- A single interface to the computing system for workload scheduling and operation of the system.
- The ability to apply all the resources of the system to a given job step when necessary.

The advantages provided by an uncoupled multiple processor configuration are:

- The capability of adding to the configuration in smaller increments, that is, the addition of a smaller processor rather than replacement of the existing processor with the next larger processor when additional computing power is required. The next larger processor may provide additional computing power far in excess of that required.
- More economical growth possibilities for installations with purchased systems.
- Growth possibilities for large-scale installations that have the largest processor of the system already installed.
- Enhancements to configuration available (better probability that a system will be available for critical

application processing), concurrent maintenance, and improved reliability (protection of critical jobs from failures in noncritical jobs by processing them in separate systems).

2. Highlights of the 3033 Processor Complex, Attached Processor Complex, and Multiprocessor Complex

The 3033 Processor Complex consists of the 3033 Processor, 3036 Console, and 3037 Power and Coolant Distribution Unit. The 3033 Processor is a high-speed, large-scale, advanced function processor of System/370. It has a significantly higher internal performance than System/370 Models 165 and 168.

The 3033 Processor is a general purpose processor and offers high performance for both commercial and scientific applications. The 3033 processor has hardware features and programming systems support, such as that for virtual storage and virtual machines, that are designed to facilitate application development and maintenance. In addition, a 3033 Processor Complex, its I/O devices, and its programming support can ease the expansion of data base and online data processing operations.

The 3033 Multiprocessor Complex is a tightly coupled multiprocessing configuration that consists of two 3033 Processors with multiprocessing hardware interconnected via the 3038 Multiprocessor Communication Unit, two 3036 Consoles, and two 3037 Power and Coolant Distribution Units.

The 3033 Attached Processor Complex is a tightly coupled multiprocessing configuration that consists of a 3033 Processor with multiprocessing hardware interconnected to a 3042 Attached Processor via the 3038 Multiprocessor Communication Unit, two 3036 Consoles, and two 3037 Power and Coolant Distribution Units. Like a 3033 Multiprocessor

Complex, the attached processor configuration can execute two instruction streams (tasks) simultaneously, one in each processor.

B. GENERAL DESCRIPTION

The 3033 Attached Processor Complex provides a growth path for 3033 uniprocessor users who require additional internal performance, but do not require all the advantages offered by a 3033 Multiprocessor Complex, and offers advantages over two uncoupled 3033 uniprocessor configurations. A 3033 Attached Processor Complex operating under OS/VS2 is capable of providing internal performance 1.6 to 1.8 times that of a 3033 Processor Complex. The internal performance improvement realized when a given 3033 uniprocessor configuration is upgraded to a 3033 attached processor configuration is dependent on the amount of multiprogramming that can be achieved. [Ref. 15]

C. COMPONENTS

The components of the 3033 Attached Processor Complex, are:

- One 3033 Model A Processor. Functionally this unit is a uniprocessor model of the 3033 Processor with tightly coupled multiprocessing hardware like that implemented in multiprocessor models of the 3033 Processor.
- One 3042 Attached Processor (AP) physically connected to the 3033 Model A Processor via one 3038 Multiprocessor Communication Unit. The 3042 AP contains an instruction processor function similar in capability to that in the 3033 Model A Processor. The 3042 does not contain any processor storage or channels. The physical size of the 3042 AP is smaller than that of the 3033 Model A Processor, since the 3042 AP does not contain channel and

processor storage frames. The 3038 provides a communication path between the two processors and the two 3036 Consoles as well as processor storage addressing capabilities for the 3033 and 3042.

- Two 3036 Consoles, one for the 3033 Processor and one for the 3042 AP. The inclusion of a 3036 Console for the 3042 AP provides independent power control and power monitoring for the 3042. This 3036 Console can also be used to execute limited diagnostics on a 3042 AP concurrently with operation of the 3033 Processor.
- Two 3037 Power and Coolant Distribution Units, each with a multiprocessing feature installed. Two motor generator sets, one for each processor, are also required.

The 3033 Attached Processor Complex can operate in two modes. When attached processor (AP) mode is in effect, the 3033 Processor and 3042 AP normally operate together as a tightly coupled multiprocessing configuration that shares processor storage in the 3033 Processor. When uniprocessor (UP) mode is in effect, the hardware connection between the 3033 and 3042 is not enabled and only the 3033 Processor can operate as a uniprocessor.

A 3033 Attached Processor Complex can be field converted to a 3033 Multiprocessor Complex. The 3033 Model A Processor must be converted to a 3033 Model M Processor and the 3042 AP must be removed and replaced with a second 3033 Model M Processor.

1. <u>3033 Model A Processor</u>

The major elements in the 3033 processor are:

- The instruction preprocessing function and execution function which form the instruction processor function that executes the instruction set for the 3033 Processor.
- Processor storage
- The processor storage control function, which controls all

access to processor storage by the other processor elements and the console and performs virtual-to-real address translation.

- Two or three channel groups and their directors.
- Maintenance and retry function.

Standard Features for the 3033 Processor are:

- Basic Control (BC) and Extended Control (EC) mode of operation and control registers.
- Instruction set that includes binary, decimal, floating-point, and extended precision floating-point arithmetic.
- Dynamic Address Translation
- reference and Change Recording
- System/370 Extended Facility
- Instruction retry.
- Interval timer (3.3 ms resolution)
- Time-of-day clock
- Clock comparator and CPU timer
- Monitoring feature
- Program Event Recording
- Expanded machine check interruption class.
- Byte-oriented operands.
- Store and fetch protection
- High-speed buffer storage - 64k bytes.
- Two channel groups, each with six channels (one byte and five block multiplexer)
- Channel indirect data addressing
- Limited channel logout area with I/O retry data and an extended channel logout.
- Reloadable control storage for the execution function and channel groups.
- Store status function
- Direct Control.

Optional features for the 3033 Processor, which can be field installed, are:

- Channel-to-Channel Adapter (one in the first group and one in the second channel group)
- Extended Channels (third channel group and director with channels 12 through 15)
- Two-Byte Interface (for block multiplexer channels 1, 7, and 12 or 13 only)

2. <u>3042 attached Processor</u>

The 3042 Attached Processor, with a 57-nanosecond cycle time, contains an IPPF, execution function, PSCF, and maintenance and retry function, like those of the 3033 Model A Processor. The 3042 AP basically differs from Model A and uniprocessor models of the 3033 Processor in that the 3042 does not contain any processor storage or channels. The PSCFs in the 3042 AP and 3033 Model A Processor communicate with each other to support the sharing of all processor storage available in the 3033 Processor. There are no optional features for the 3042 AP.

3. 3038 Multiprocessor Communication Unit

The 3038 Multiprocessor Communication Unit physically connects the two processors in a 3033 Attached Processor Complex. It provides a communication path between the two processors and between the two 3036 Consoles. Functionally, the 3038 is divided in half. Each half is associated with the processor it is attached to and receives its power and water cooling from the 3037 of its associated processor. Each half of the 3038 can be powered up and down separately from the other half and contains and oscillator for timing its associated processor.

4. 3036 Consoles

The two 3036 consoles are used to perform the following major functions:

- Sequence, monitor, and control power.
- Load instruction processor (execution function), director, and console microcode
- Configure certain processor components and 3036 Console functions.
- Control 3033 Processor operations (IPL, start, stop, etc.) and communicate with the operating system being used.
- Configure the 3036 Console for concurrent maintenance operations.
- Exercise processor components to locate a malfunction.
- Display processor and channel indicators and logouts
- Execute microdiagnostics for the 3036 and 3033 that are contained on diskettes.
- Perform remote maintenance operations.

The two 3036 Consoles in a 3033 Attached Processor Complex are Physically identical to the 3036 Console used in a 3033 Processor Complex. No changes are made to the

control panel or the two operating stations. However, the operational diskettes for a 3033 attached processor configuration contain processor and console microcode that is designed to support multiprocessing and interprocessor communication.

The 3036 Console associated with the 3042 AP is cable connected to the 3042. This cabling provides certain hardwired communication functions between the 3042 AP and its console such as powering, logging to the diskette, hardware configuring using a configuration display, indicator displays, etc.

Operator to operating system communication utilizing a display station, program access to a diskette drive operating in service record file mode and remote maintenance operations can also be performed using the 3036 Console associated with the 3042 AP. These operations require the 3036 console to be attached to a channel in the 3033 Processor, as is the other 3036 Console, and assigned three I/O addresses.

For attached processor mode operations at least one operating station in each 3036 Console must be functional. This is necessary for power control for each processor and 3036 Console to 3036 Console communication which is accomplished via the service support stations. All logouts from a processor are written to the diskette drives of its associated 3036 Console.

In the normal console configuration for attached processor mode operations, one station in each 3036 Console is designated as an operator station while the other is designated as a service support station. One operator station in either 3036 Console is used as the primary operating system console while the other can be defined as a alternate or additional console.

D. ADVANTAGES OF A 3033 ATTACHED PROCESSOR CONFIGURATION.

In addition to increased internal performance over a uniprocessor configuration, an attached processor configuration offers advantages over two uncoupled uniprocessor configurations with the same total resources as the attached processor configuration. [Ref.16]

1. Less complex Operational Requirements

An attached processor configuration has less complex operational requirements than two uncoupled systems because it presents a single system image to the operator even though there are two instruction processor functions in the configuration. The operator has one operational interface to the entire system, one job scheduling interface, and one point of control for all the resources in the configuration. In addition, the operator must communicate with and control only one control program instead of two.

2. Improved Resource Utilization

Resource utilization in an attached processor configuration is improved over that of two uncoupled systems because load leveling occurs between the two systems, there is a reduction in the amount of processor storage required by the resident control program, all I/O devices in the configuration can be accessed by the 3033 Processor, and the need for using shared DASD support is eliminated.

Load leveling occurs for the two processors because of the way in which VM/370 can schedule task execution in a tightly coupled configuration. Load leveling reduces the peak and valley periods of processor utilization that normally occur in two uncoupled systems, as follows.

The two processors are considered to be system resources that, when available, are allocated to ready tasks. Usually, either processor is capable of processing each task in the system. Thus, as soon as a processor becomes available, it is allocated to the highest priority queued ready task. Since there are on the average twice as many tasks in an attached processor configuration than in one system in a two-uniprocessor environment, the chances are significantly reduced that no task in the attached processor configuration will be ready to execute and hence available processor time will be unutilized.

Since there is only one copy of the VM/370 multiprocessing control program resident in processor storage in an attached processor configuration, more processor storage is available for paging (which can benefit performance) than in two uncoupled systems with the same total amount of processor storage as in the Model A 3033 Processor, each of which has an VM/370 uniprocessor control program resident.

While the 3042 AP cannot issue I/O instructions, it can process data read and to be written by the 3033 Processor. Therefore, in effect, the I/O devices in a 3033 attached processor configuration are pooled for use by both processors. More than half the total number of I/O devices present can be allocated to an individual job step when necessary. The pooling of I/O devices and sharing of processor storage permits the execution of jobs with larger processor storage and I/O device requirements than can be handled using one system in a configuration with two uncoupled systems.

The sharing of processor storage and the ability of the 3033 Processor to access all I/O devices in the configuration also enables the VM/370 control program to

automatically handle peak load situations within jobs and to balance the processing across the two processors. Manual balancing of the workload between two systems, as is required for two uncoupled systems, is not required for a tightly coupled configuration.

Through pooling the number of I/O devices in a tightly coupled multiprocessing configuration can be less than the number of I/O devices needed for two uncoupled systems that are to handle the same large I/O job or peak load direct access storage requirements.

Since there is only one control program for an attached processor configuration, there is no need to split any data base into two parts, one for each system, or to use Shared DASD support in order to share a data base between the two systems. The use of Shared DASD support results in reduced throughput for two uncoupled systems because of the interference it introduces. This throughput reduction is not incurred in an attached processor configuration since there is only one OS/VS2 MVS control program and it can maintain the integrity of a shared data base without using Shared DASD support.

E. NPS IBM 3033 System Configuration

The configuration of the IBM 3033 AP System installed in the W.R. Church Computer Center, Naval Postgraduate School, is the following:

IBM 3033 AP SYSTEM

QUANTITY		ITEM
CESSORS		
1	IBM 3033-A16	Central Processing Unit (16 Mbytes)
1	3042-1	Attached Processor Unit
2	3036-1	Console (2 CRTs each)
2	3037-1	Power and Coolant Distribution Unit
1	3038-1	Multi-processor Communications Unit
MS		
2	IBM 2835-2	Drum Storage Control
2	2305-2	Drum Storage (12 Mbytes each)
K		
2	IBM 3830-2	Disk Storage Control
2	3350-A2	Disk Storage & Control (2 spindles @ 317.5 Mbytes each)
4	3350-A2F	Disk Storage (Fixed head) (2 spindles @ 317.5 Mbytes each)
6	3350-B2	Disk storage (2 spindles @ 317.5 Mbytes each)
2	3350-B2F	Disk Storage (fixed head) (2 spindles @ 317.5 Mbytes each)
1	*ITEL 7830-1	Disk Storage Controller
3	*ITEL 7330-1	Disk Storage (100 Mbytes each)
1	* 2314-1	Disk Storage (8 spindles @ 29 Mbytes each)
S STORAGE SYSTEM (MSS)		
1	IBM 3851-A2	Mass Storage Facility (cartridge, 50 Mbytes ea.)
2	3830-3	Disk Storage Control
2	3350-A2	Disk Storage & Control (2@100 Mbytes each)
2	3350-B2	Disk Storage
ES		
2	* IBM 3803-1	Tape Control
6	* 3420-8	Tape Drives (6250 bpi)
1	* 2803-1	Tape Control
1	* 2402-1	Tape Units (2 drives)
MINALS - COMMUNICATIONS		

5	IBM	3274-D1	Control Unit (32 Terminals each)
31		3278-2	Display (EBCDIC)
25		3278-2	Display (APL/EBCDIC)
6		3262-3	Printer (3278-compatible)
1		3777-1	Communications Terminal
1		3203-3	Line Printer (remote)
1	*	3705	Communications Controller

Lines:			
	@9600	bps	RJE
4 dial-up	@1200	bps	ASCII
8 " "	@ 300	bps	"
6 hard-wired	@ 300	bps	"
1 " "	@4800		IBM 3276
1 " "	@9600		" "
2 " "	@9600		RJE

MINALS-COMMUNICATIONS

1	* IBM	2702-1	Transmission Control
		LINES: 4 dial-up @134.5 bps; 4 hard-wired @134.5 bps	
1	* IBM	2701	Data Adapter Unit
		LINES: 3 hard-wired @4800 bps	

UT/OUTPUT DEVICES

1	* IBM	2821-1	Control Unit
1	*	2821-2	Control Unit
2	*	1403-N1	Printer (1000 lpm)
1	*	2540-1	Card/Reader/Punch
1	*	2501-B2	Card Reader (self-service)
1	*VERS	8222-A	Electrostatic Printer/Plotter

TES: 1. All equipment is IBM unless otherwise indicated.
2. The asterisks indicate the deviceds retained from old system.

IBM 3033 SYSTEM CONFIGURATION

	0	2701	3705 (Communications)	2702	3851 Mass Storage	3036 CPU Console	3036* AP	2821 1403 2540 2821 1403
CHANNELS (Group 0)	1	2835 2305	2835 2305					Drums
	2	3830-3 (2) 3350	ITEL 7830 (3) 7330-1	3830-3 (2) 3350				Disk
3042-1	3	3830-2 (6) 3350	3830-2 (6) 3350					Disk
ATTACHED PROCESSOR	4	3274D1 (26) 3278	3274D1 (24) 3278					Terminal
3038 MCU	5	2803 (1) 2402	3803* (1) 3420-7 (2) 3420-8	3803 (3) 3420-8	2314 (8)			Tapes, Disks
3033 CENTRAL	6	3851* MSS	2501 Reader (Hot)	2702* TCU	3705* CC 3036 AP Console	VERS PLOT 3036* CPU		2821* 1403 2540
PROCESSOR UNIT	7	2835* 2305	2835* 2305					Drums
	8	3830-3 (2) 3350 (3) 7330-1	ITEL* 7830-1	3830-3 (2) 3350				Disk
CHANNELS (GROUP 1)	9	3830-2 (6) 3350	3830-2* (6) 3350					Disk
	A	3274D1 (26) 3278	3274D1 (24) 3278	3274D1 (24) 3278				Terminals
	B	2803* (1) 2402 (2) 3420-8	3803* (1) 3420-7	3803* (3) 3420-8	2314* (8)			Tapes, Disks

V. VM/SP PERFORMANCE MEASUREMENT TOOLS

A. INTRODUCTION

Measurement facilities, broadly speaking, have been implemented at several levels of the Virtual Machine Facility/370. The user level, the system operator level, the system analyst level, and the installation management level.

Three commands [Ref.17] INDICATE, QUERY SRM, and MONITOR, provide a way to dynamically measure system performance.

Indicate: provides the system analyst and general user with a method to observe the load conditions on the system while it is running.

QUERY SRM: Provides the system operator with expanded observation facilities for analyzing internal activity counters and parameters.

MONITOR: Provides the system analyst and the system operator with a data collection tool designed for sampling and recording a wide range of data. The collection of data is divided into functional classes. The different data collection functions can be performed separately or concurrently. Keywords in the Monitor command enable the collection of data and identify the various data collection classes. Other keywords control the recording of collected data or tape for later examination and reduction.

The VM/370 performance measurement tools are three separate entities [Ref.18]. The first known as LOAD INDICATORS includes the INDICATE and QUERY SRM commands. It's purpose is to provide the users, analysts, and the operator with the means to observe the load conditions on the system, to varying degrees depending on necessity and the levels of authority granted them, such that they may base their respective dealings with the system on data

obtained in real-time. In the case of performance problems, real-time data may be of limited usefulness and may only serve to confirm that a problem exists. There may be no way of understanding, from a small amount of on line data, the exact nature of the problem, how it arose, or how it may be resolved.

The second, known as VM/SP Monitor, is a general purpose control program service for collecting a wide range of data relating to most aspects of performance measurements. This service collects data on tape for later reduction and analysis. In most cases, data collection takes place with insignificant overhead and, thus, minimal impact on the system being monitored. (Reduction of the data may, of course, constitute a significant load but may be done with batch facilities).

The third service is an optional data reduction system known as Statistics Generating package for VM/370, VM/SGP. It is available as an Installed User Program. VM/SGP consists of a data selection and reporting language, a translator, and a library of reduction programs to handle most classes of VM/SP Monitor output.

Therefore, together with a reduction facility, permanent records of summarized measurements of load and performance may be maintained, and problems may be thoroughly analyzed and resolved with definitive results. The great flexibility of VM/SGP may be employed to massage the raw data into forms most acceptable by management or which carry the greatest impact for the given circumstances.

B. THE INDICATE COMMAND

The indicate command [Ref.17] allows the general user and the system analyst to display at their consoles at any time, the usage of and contention for major system resources.

The general user can display usage of and contention for the major system resources of processor and storage. He can also display the total amcunt of resources he has used during his terminal session and the number of I/O requests. If he uses the Indicate command before and after the execution of a program, he can determine the execution characteristics of that program in terms of resource usage.

The system analyst can identify active users, the queues they are using, their I/O activity, their paging activity, and many other user characteristics and usage data.

The VM/SP Schedule maintains exponentially smoothed values for data provided by the LOAD option. Specifically, at intervals (in seconds) depending on the processor model, the scheduller calculates the total activities for variables such as CP and storage usage for the most recent interval, and factors them into a smoothed wait value in the following way:[Ref.19]

$$NSWV = \frac{(3 \times \text{old smoothed wait value} + \text{current interval wait})}{4}$$

Where NSWV = New Smoothed wait value

Thus, only 1/4 of the most recent interval wait is factored into the new smoothed wait which makes it predominantly the cld smoothed wait value.

The remaining INDICATE components are sampled prior to a user being dropped from a queue. Because of the frequency of this event, the remaining components are subject to a heavier smoothing than the wait time. A general expression for the smoothing follows:

nsv = ((rate - int) (osv)/rate) + civ

<u>where</u>:

nsv = new smoothing value
csv = old smoothing value

civ = current interval value (results found during the current interval (int)

int = current interval (time period being tested)

rate = either history interval (hrate) of 8 minutes, or data interval

Other operands of the command allow users to obtain other performance information that enables them to understand the reasons for the observed conditions.

<u>The Class G Indicate Command</u>

The format of the class G (general User) Indicate command is:

```
INDICATE        LOAD
                -----
                USER
                -----
```

Where:

Indicate LOAD produces the following response, where n is a decimal number [Ref.20]

```
CPU-nnn%     APU-nnn%     Q1-nn Q2-nn STORAGE-nn% EXPAN-nnn
PROC xx-nn%  PROC YY nn%
PAGING-nnn/sec, STEAL-nnn%,LOAD-nnn%
```

The CPU figure indicates the percentage of time that the main processor is running and is derived from the smoothed wait value maintained by the scheduler. In a multiprocessor environment, PROC xx-nnn% is a smoothed value that indicates the percentage of time that the system is running on the IPL processor.

The APU figure is the percentage of time the attached processor is running. In a multiprocessor environment, PROC YY-nnn% is a smoothed value that indicates the percentage of time that the system is running on the non-IPL processor.

The contention for the processor is represented by

average numbers of users in queue 1 and queue 2, maintained by the scheduler.

The next field, STORAGE, is a measure of the usage of real storage. It is an approximation of the number of pages in storage for in-queue virtual machines to the number of pageable pages in the system, expressed as a percentage.

Due to the algorithm used by the scheduler in approximating the number of pages in storage, the value of STORAGE can exceed 100%.

The value may also be less than 100% even when the sum of the estimated working set for in-queue virtual machines is greater than the number of pageable pages.

The scheduler contention value, EXPAN, is the smoothed measure of the time it takes a virtual machine to receive a given amount of processor time, as follows:

$$EXPAN = \frac{QT}{Q2\ slice}$$

Where:

QT is the average elapsed time between queue drops a Q2 virtual machine spends in the Q2 dispatch list.

Q2 slice is the amount of processor time allowed to a Q2 virtual machine in the dispatch list.

EXPAN is a measure of the total delay in response time that a virtual machine experiences due to contention for both real storage and the processor. It is maintained at a minimum value of 1.

PAGING is the average number of page I/O operations (page reads and writes) performed per second.

STEAL is the percentage of page read operations that required a real storage page to be stolen from another in-queue virtual machine.

LOAD is an artificial value that attempts to measure (in terms of elapsed time) the percentage of the system devoted to paging because of real storage contention. Estimated processor time involved in paging, the amount of time spent in pagewait, and the percentage of steals are factored into the calculation.

INDICATE USER

Allows a user to determine the resources used and occupied by his virtual machine, and the I/O events that have taken, place.

The following two line response is returned:

PAGES: RES-nnn WS-nnn READS=nnn WRITES=nnn MH-nnn PH-nnn
VTIME=nnn:nn TTIME=nnn:nn SIO=nnn RDR-nnn PRT-nnn PCH-nnn.

The first line of the response displays the data from the user's VMBLOK that is relevant to his virtual machine's paging activity and resource occupancy.

RES: is the current number of the user's virtual storage pages resident in real storage at the time the command is issued.

WS: is the most recent system estimate of the user's working set size.

READS: is the total number of page reads for this user since he logged on or since the last ACNT command was issued for his virtual machine.

WRITES: is the total number of page writes for this user

since he logged on or since the last ACNT command was issued for his virtual machine.

MH: is the current number of virtual pages allocated on the system moveable head preferred paging area for this user.

FH: is the current number of virtual pages allocated on the system fixed head preferred paging area for this user.

The second line of the response gives the user his processor usage and accumulated I/O activity counts since logon or since the last ACNT command was issued for his virtual machine.

VTIME is the total virtual processor time for the user.

TTIME is the total virtual processor and simulation time for the user.

SIO is the total number of nonspooled I/O requests issued by the user.

RDR is the total number of virtual cards read.

PRT is the total number of virtual lines printed.

PCH is the total number of virtual cards punched

THE CLASS E(SYSTEM ANALYST) INDICATE COMMAND

The format of the class E Indicate command is:

```
INDicate  | |LOAD                    |
          | |USER       |*       | |
          | |           |-       | |
          | |           |userid  | |
          | |Queues                  |
          | |I/O                     |
          | |PAGing     |WAIT    | |
          | |           |ALL     | |
```

Where:

INDICATE LOAD

provides the same output as the INDICATE LOAD cption described under "The Class G Indicate Command."

INDICATE USER *

reflects activiy cf the system analyst's own virtual machine. The output of this option is the same as that of the INDICATE USER * option described under "The Class G INDICATE Command".

INDICATE USER USERID

allows the system analyst to determine the activity of other virtual machines in terms of the resources used and cccupied and events that have taken place. Users with class E authcrity can access data from the VMBLOK of any user currently lcgged onto the system in their attempts tc understand an overload or pcor performance situation.

INDICATE QUEUES

displays the active users, the queues they are in, the storage they are cccupying, and the status they are in. The display indicates those users currently dominating main storage. Users waiting in eligible lists are included in the respcnse because they are contending for main storage and it is only by chance that they were not occupying main stcrage at the lime of the command.

INDICATE I/O

provides information about conditions leading to possible I/O contenticn within the system. The response gives the userids of all the users in I/O wait state at that instant in time, and the address of the real device to which the mcst recent virtual SIC was mapped. Because the respcnse indicates only an instantaneous sample, use the command several times before assuming a

condition to be persistent. If it is persistent, run the SEEKS option of the MONITOR command to conduct a thorough investigation of the suggested condition.

INDICATE PAGING WAIT
is provided for installations that have 2305s as primary paging devices and other direct access devices as secondary paging devices. A full primary device and subsequent allocation of paging space on the slower device may be responsible for degradation in system performance. Use the INDICATE PAGING WAIT option when the INDICATE QUEUES option shows that a significant proportion of the users in queue1 and queue2 are persistently in page wait. The response to the command gives the userids of those users currently in page wait and the numbers of page frames allocated on drum and on disk.

INDICATE PAGING ALL
displays the page residency data of all users of the system (including the system nucleus and pageable routines). The response is identical to that of the INDICATE PAGING WAIT option.

C. QUERY SRM AND QUERY PAGING COMMANDS

1. <u>Querying and Setting the System Resource Management Variables</u>

The QUERY SRM and SET SRM commands allow the system analyst to query and/or change internal system activity counters or parameter.

The system analyst can use the class E QUERY SRM command to display the following information:

- current number of pageable pages
- size of the dispatching time slice

- Setting of the maximum working set estimate
- Maximum drum page allocation limit
- Current page migration counters
- Unused segment elapsed time as criteria for page migration.
- Current PCI flag setting mode for 2305 page requests
- Maximum page bias value
- Current interactive shift bias value
- Moveable head page migration limit.

The class E SET SRM command allows the system analyst to set some of the system variables that can affect the values displayed by the QUERY SRM command.

2. Querying and Setting the Paging Variable

The paging variable is used in the working set size algorithm. The current paging load is constantly compared with the paging variable. Adjustments are then made in the working set size estimates, based on how well the actual load compares with the paging load variable.

The QUERY PAGING command displays the paging variable used in the working set size estimate control algorithm. Information on the paging rate per second is available as a response to the INDICATE LOAD command.

The SET PAGING command is used to change the paging variable used in the working set size estimate.

D. THE MONITOR COMMAND

VM/SP Monitor collects data in two ways:

1. By handling interruptions caused by executing MONITOR CALL (MC) instructions.

2. By using timer interruptions to give control periodically to sampling routines.

MONITOR CALL instructions with appropriate classes and codes are presently embedded in strategic places throughout the main body of VM/SP code (CP). When a MONITOR CALL instruction executes, a program interruption occurs if the particular class of MONITOR CALL is enabled. The classes of MONITOR CALL that are enabled are determined by the mask in Control Register 8.

When a MONITOR CALL interruption occurs, the CP program interruption handler transfers control to the VM/SP Monitor interruption handler where data collection takes place.

Sixteen classes of separately enabled MONITOR CALL instructions are possible, but only eight are implemented in the VM/SP Monitor.

Monitor output consists of event data and sampled data. Event data is obtained via MONITOR CALL instructions placed within the VM/SP code. Sampled data is collected following timer interruptions. All data is recorded as though it were obtained through a MONITOR CALL instruction. This simplifies the identification of the records.

The MONITOR command:

- Stops and starts CP internal trace table data collection.

- Displays the status of the internal trace table and each implemented class of VM/SP Monitor data collection. In addition, it displays those specifications for automatic monitoring that are overridden by Monitor commands. It also displays whether the tape, or spool file is the recording medium.

- Starts and stops VM/SP data collection using tape or spool file. It also closes the spool file, if desired.

- Specifies VM/SP monitor classes of data collection enabled, number of buffers used, and time of data

collection. It also specifies other options which override the specifications for automatic monitoring contained in DMKSYS.

- Specifies the interval to be used for timer driven data collection.

- Specifies direct access devices that are to be included or excluded from a list of devices. The list defines direct access devices for which CP is to collect data for the SEEKs class.

NOTE: for more detail information about the format of the Class A and E Monitor command, see [Ref.21]

E. ACCOUNTING RECORDS

The accounting data gathered by VM/SP can help in analysis of overall system operation. Also, accounting data can be used to bill VM/SP users for time and other system resources they use.

There are three types of accounting records [Ref.22]. The virtual machine user records, records for dedicated devices as well as T-Disk space assigned to virtual machine users, and accounting records generated as a result of user initiated DIAGNOSE X'4C' instruction.

This service is only given to users with the account option (ACCT) in his directory.

1. Accounting Records for Virtual Machine Resource Usage

The information stored in the accounting record in card image form when a user ends his terminal session (or when the ACNT command is invoked) is as follows. (Columns 1-28 contain character data, all other data is in hexadecimal form, except as noted

Collumn	Contents
1- 8	Userid
9-16	Account number
17-28	Date and Time of Accounting (mmddyyhhmmss)
29-32	Number of seconds connected tc VM/SP System
33-36	Milliseconds of processor time used, including time for VM/SP supervisor functions.
37-40	Milliseconds of virtual processor time used
41-44	Number of page reads
45-48	Number of page writes
49-52	Number of virtual machine SIO instructions for nonspooled I/O
53-56	Number of spool cards to virtual punch
57-60	Number of spool lines to virtual printer (this includes one line for each carriage control command)
61-64	Number of spool cards form virtual reader
65-78	Reserved
79-80	Accounting record identification code (01)

2. <u>Accounting records for dedicated devices and temporary disk space</u>

Accounting records are recorded and spooled to disk when a previously dedicated device and temporary disk space is released by a user via DETACH, LOGOFF, or releasing from DIAL (dedicated device only). A dedicated device is any device assigned to a virtual machine for that machine's exclusive use. These include devices dedicated by the ATTACH command, those being assigned at logon by directory entries, or by a user establishing a connection (Via DIAL) with a system that has virtual 2702 or 2703 lines. The information on the accounting record is card image form is as follows

Column	Contents
1- 8	Userid
9-16	Account number
17-28	Date and time of Accounting (mmddyyhhmmss)
29-32	Number of seconds connected to VM/SP system
33	Device class
34	Device type
35	Model (if any)
36	Feature (if any)
37-38	Number of cylinders of temporary disk space used (if any) or number of blocks used (columns 37-40) for fixed-block devices. This information appears only in a code 03 accounting record.
39-78	Unused (columns 41-78 unused for fixed-block devices)
79-80	Accounting record identification code (02,03)

3. Accounting Records of Logon, Autolog, and Link Journaling

When LOGON, AUTOLOG, and LINK journaling is ON, VM/SP may write type 04,type 05, or type 06 records to the accounting data set. These records are written when VM/SP detects that a user has issued enough LOGON or AUTOLOG commands with an invalid password to reach or exceed an installation defined threshold value; or when a user has successfully issued a LINK command to a protected minidisk not owned by that user; or when a user has issued enough LINK commands with an invalid password to reach or exceed an installation defined threshold value.

4. Accounting Records Created by the User

A virtual machine can initiate the creation of an accounting record that contains up to 70 bytes of information of his own choosing. To do this, he issues a DIAGNOSE code X'4C' instruction with the following operands:

- The address of a data Area in virtual storage containing the information in the actual format, that he wishes to have recorded in columns 9 through 78 of the card image record.
- A hexadecimal function code of X'10'
- The length of the data area in bytes.

The information on the accounting record is as follows:

Column	Contents
1- 8	Userid
9-78	User formatted data
79-80	Accounting Record identification code(CO)

For details on DIAGNOSE instructions in a Virtual Machine see [Ref.23].

VI. PERFORMANCE ANALYSIS OF THE VM/370

A. INTRODUCTION

The performance of VM/370 systems is analyzed with the objective of developing performance and resource usage forecasting equations. A model is developed for analyzing computer performance and resource allocation, for computer systems using this operating system.

In this chapter we will present a methodology for doing the analysis, showing how response times and resource utilization are related to the system load.

Some performance measurement and analysis techniques [Ref.24,25] emphasized the detection of performance improvements caused by changes in the system. Here, no attempt was made to analyze the causes for improvement or lack of it, nor to relate these to the internal structure of the system. What we shall try to accomplish is to relate the trends in the data to the performance of the system and thus gain an insight into what might cause the system to saturate and its performance to degrade.

B. SYSTEM DESCRIPTION

For a description of the VM/SP (VM/370) see Chapter III. For the configuration of the IBM 3033 AP, used in the W.R. Church Computer Center NPS, see Chapter IV.

For the analysis purposes, the VM/CMS is simply a virtual-machine time-sharing system. Each user of the system may enter tasks, normally from a remote terminal. The system shares its resources among these tasks. The flow of user tasks through the system is depicted in Figure 1. [Ref.26]

A user is in the "dormant" state until he has completed creating a task. Until proven otherwise, the task is assumed to be "interactive", i.e. to require fast response while making only slight demands on system resources. While receiving service, such tasks are said to be in "Q1", but before being admitted to this state they are called "Q1 Candidates". If a Q1 task does not terminate before consuming a certain amount of CPU time (roughly 400 milliseconds), it loses its "interactive" status. It now becomes a "Q2 Candidate" and is elegible to be admitted into "Q2", which is the set of noninteractive tasks currently being serviced. There is also a limit (about five seconds) on the amount of CPU time that a task may receive during one stay in Q2. A task requiring more CPU time may cycle several times between the "Q2 Candidate" and "Q2" states.

Not all the possible state transitions are shown, but those shown are enough for our purposes.

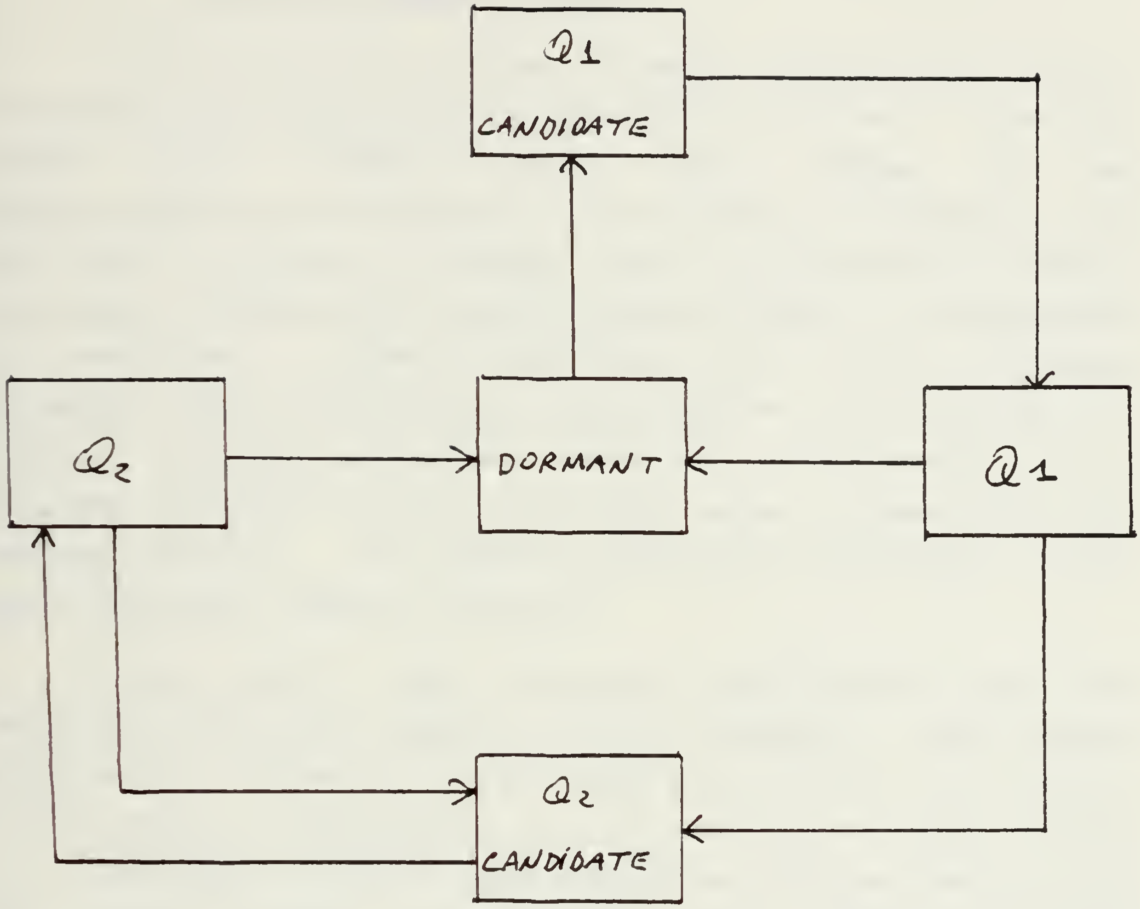

FIG. 1 USER STATES AND PRINCIPAL TRANSITIONS

C. DEFINITIONS AND TYPICAL RELATIONS

1. Basic Definitions

a. Multiprogramming Level

The only tasks that may actually receive CPU time at any moment are those in the Q1 and Q2 states, those are called 'in-Q' tasks, and their number is the multiprogramming level (MPL). The Q1 and Q2 candidates are tasks that are ready to run but are not allowed to do so at the moment because the system does not wish to overcommit its resources. These tasks are said to be eligible. Admissions from eligible to in-Q status is in order of deadline priority which is based on a combination of a permanent directory entry, the time when the user had last received service, and optional penalties depending on the user's previous resource demands.

In-Q user's main storage requirements are met dynamically through a demand paging mechanism. The system maintains an estimate of each user's storage requirements; this estimate is referred to as the user's projected working set. Admission is based principally on the availability of main storage space to accommodate the user's projected working set.

While a task is in Q it requires system resources, e.g. main storage, file I/O, and paging I/O, in addition to CPU cycles. The rates at which these resources are being utilized thus provide additional performance measures, as do the response times to tasks of various types. A true picture of system performance can be gained only if all these factors are considered simultaneously.

b. System Scheduler

The main function of the system scheduler is to

maintain the optimal MPL for the given system configuration and user workload. The scheduler performs its task by estimating each user's working set, and by admitting into Q only as many users whose working sets can be accommodated in main storage.

2. Important Relations

a. MPL and Performance

As the MPL goes up, the system is increasingly able to overlap the use of its various components, consequently improving their utilization. Soon, however, one or more components approach 100% utilization, so that no further increase is possible. The system is then said to be saturated. There is no benefit in increasing the MPL beyond the saturation point, which is determined primarily by the system configuration See figure 2. The saturation point is also affected by the nature of the work load if the work is unevenly spread among the system components, saturation will occur sooner than if the load is well-balanced.

At any rate, if the effects of paging are ignored, then curve A in Figure 3 shows a typical relationship between performance (as measured by CPU utilization) and the MPL. This curve might be an idealized representation of a VM/370 system with essentially infinite main storage capacity (hence, negligible paging activity).

In a real system, main storage capacity is finite. As the MPL increases, the amount of storage available to each program decreases, hence the paging rate increases. This increase becomes drastic when the storage allocated to each program falls below its "working set". Soon the paging channel becomes saturated, and further increases in the relative paging rate force down the productive utilization of other components, such as the CPU.

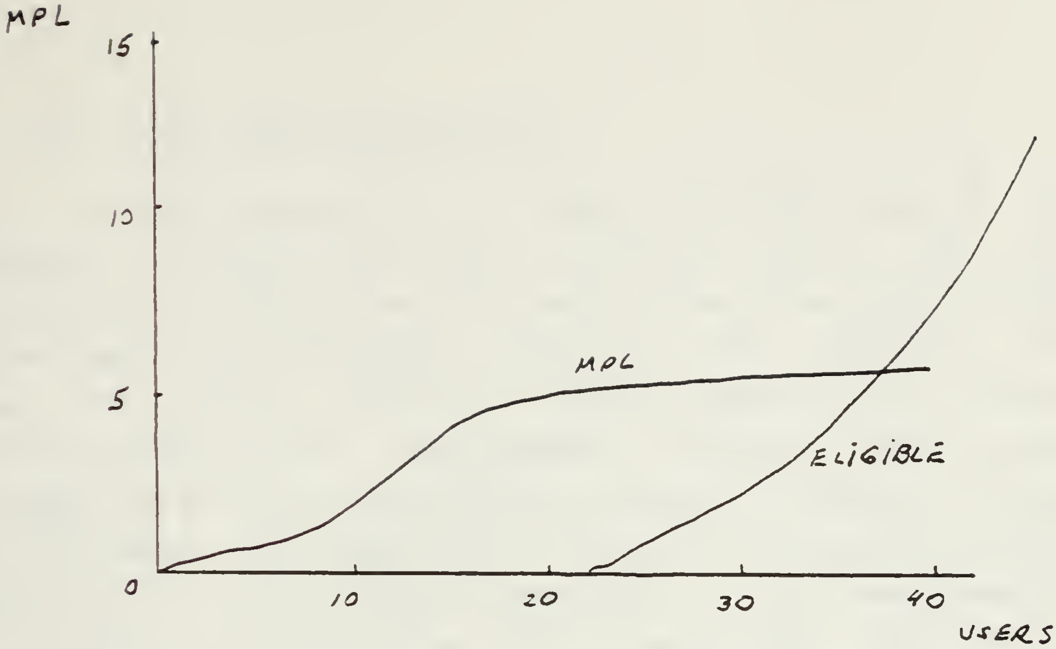

FIGURE. 2

RELATION BETWEEN MPL AND NUMBER OF USERS

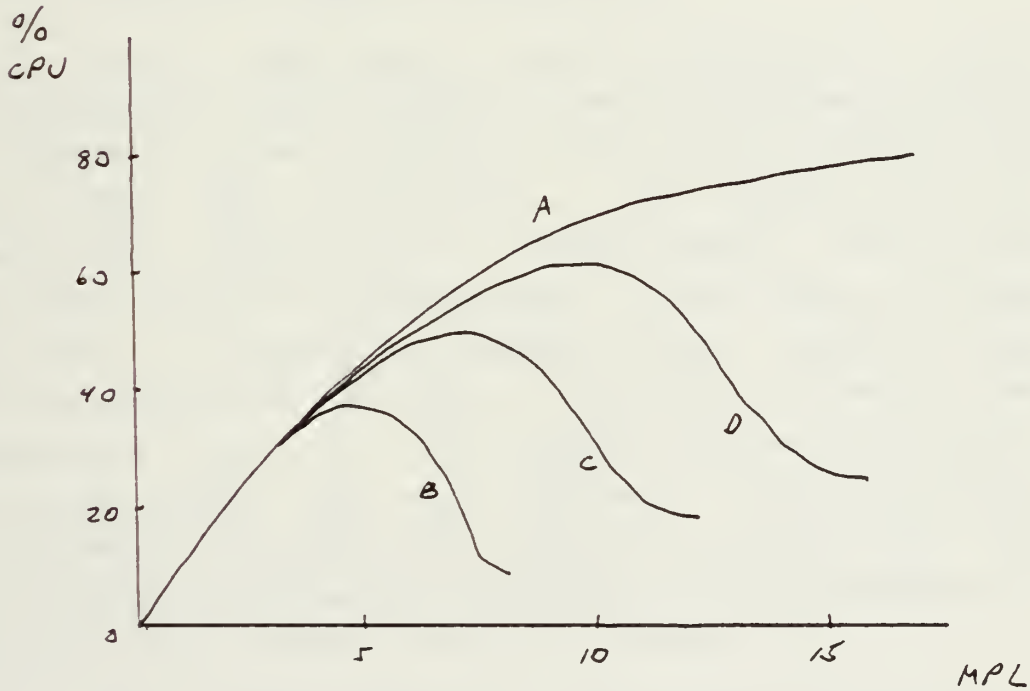

FIGURE. 3

TYPICAL RELATION BETWEEN CPU UTILIZATION AND MPL

b. Load and Performance

BARD [Ref.27], of IBM, believes that it is important to understand the relation between MPL and performance, but what one is really interested in, is the relation between performance and the load placed on the system. He took the number of active users as the primary measure of system load. The measure is open to much valid criticism. No two users are alike; how can we compare a user who is editing a file under CMS to one who is using the IPL procedure for a large virtual Machine ? yet measurements taken at many installations have shown that performance variables averaged over reasonable periods of time (one week, say) present very consistent patterns when plotted against number of active users.

c. Response Time and System Load

As the Bard Model [Ref.28] indicates, we can also determine how task response time is affected by system load. Suppose we have a specific task in mind requiring one second of CPU time to complete. The response time is composed of the running time, and the waiting time in the eligible set, which is proportional to the number of tasks in that set. To obtain the running time, suppose a task requires one second of CPU time. Let U(n) be the CPU time utilization at MPL = n. If system resources are shared fairly among users, the task receives, on the average, u(n)/n seconds of CPU TIME per second of elapsed time, and hence its running time may be estimated as n/u(n) seconds to complete the task required time of 1 second.

The proportionality factor (waiting time in eligible list Vs Number of tasks in that list) is determined: as follows If the average task required T seconds of CPU time, and the MPL = M, then a task leaves Q1

or Q2 on the average of every T/u(n) seconds. Hence, if there are M eligible tasks, the expected waiting time is M T/u(n). Thus, the relation of Figure 4 can be obtained, where the total response time is shown VS the number of users. Figure 4 shows the essential effect of saturation on response time.

This analysis applies to a typical task, i.e. one whose various resource requirements are roughly proportional to those of the overall work load. Radically different types of tasks, e.g., compute-bound or I/O bound ones, may present completely different response profiles. A system whose I/O channels are saturated may have enough leftover CPU power to maintain excellent response to a compute-bound task, and vice versa.

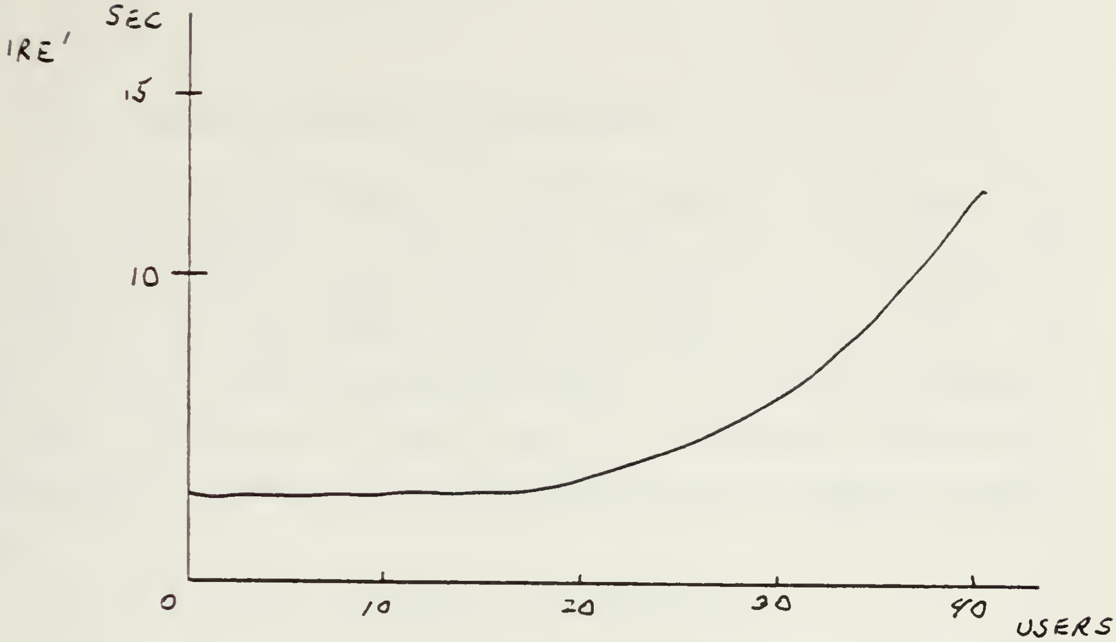

FIGURE. 4

RELATION BETWEEN RESPONSE TIME AND USER LOAD

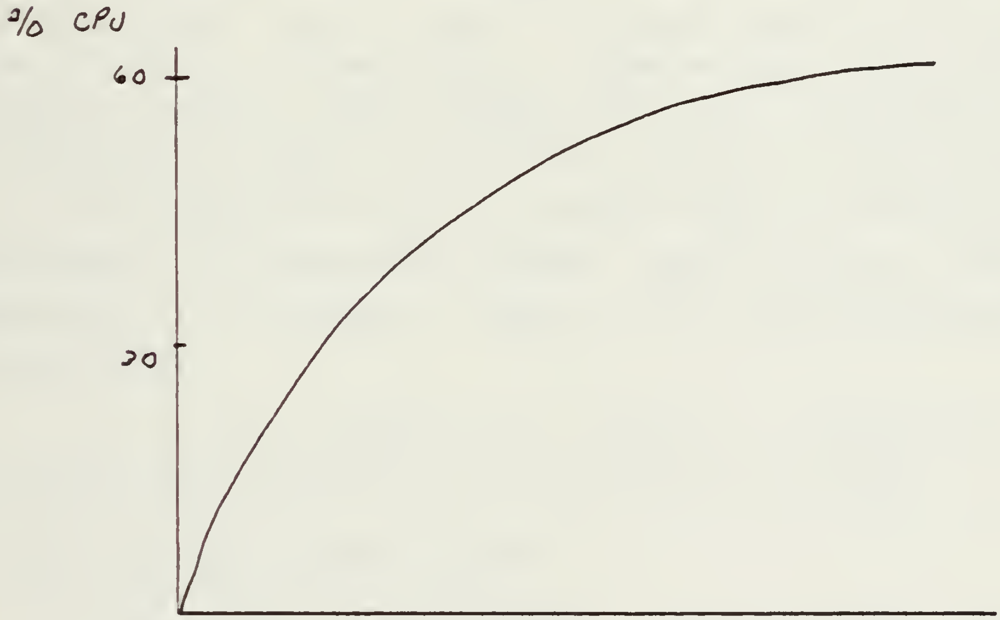

FIGURE. 5

RELATION BETWEEN CPU UTILIZATION AND USER LOAD

d. System Resource Utilization

For each number of active users, we could find the MPL (Figure 2), and then based on it find the utilization. The result is shown in Figure 5: utilization climbs up to the saturation point and then levels off. In practice, utilization actually starts to decrease somewhat beyond the saturation point, due to increased contention and increased consumption of storage space by system control blocks.

e. The effect of Storage Capacities

The typical results for various storage capacities are shown in Figure 3 [Ref.25]. In curve B, storage capacity can accommodate the working sets of three programs. Hence, utilization peaks at an MPL of three. Similarly, curves C and D depict storage capacities of six and nine programs, respectively. Observe that the maximum utilization in curve B is considerably below the infinite-storage saturation level, but is very close to it in curves C and D. It appears that only a limited gain in performance is attainable by increasing storage capacity much beyond accommodating six programs, which is the infinite-storage saturation point.

In the NPS 3033 AP the storage is fixed and we could not use this effect in this particular analysis.

f. System Expansion Factor

Typically, as the number of simultaneous users increases, the response time steadily degrades, but sharply degrades after some critical number of users. Thus, if we act so as to maximize the number of simultaneous users, the effect is to maximize the number of people to whom we are presenting a sharply degraded service. This means that the

number of simultaneous system users could be a misleading measure of system load. System expansion factor rather than number of simultaneous users could be a better measure for the computing center and for the users. This factor is defined as the ratio of the actual time to do a unit of computer-limited work to the minimum time to do that work in a stand-alone environment, and is calculated by dividing the total system service time plus queueing time to perform some user-requested action by the CPU time plus the estimated I/O time This assumes that computation time and I/O time for any one user are not overlapped. It also assumes that all paging and disk arm movement are overhead operations, induced by contention. Although this is not entirely true in all cases, it serves well enough as a first-order aproximation of what actually happens.

3. Bottlenecks

Plots such as Fig. 4 and Fig. 5 could be used to spot saturation conditions and determine the user load under which they occur. If there is no evidence of saturation, all is well. However, if saturation is detected, we must next determine the system component that is causing the saturation. This component, which is called a bottleneck, is one whose capacity must be increased first before overall system performance can be improved. Such improvement would be felt in two ways: an increased number of users could be served before the onset of saturation, and system throughput at saturation would be increased.

The main hardware components of a VM/370 system are the CPU, main storage, the paging subsystem and the I/O (other than paging subsystem). Bottlenecks in any of these components can be spotted. However, this kind of analysis is not the main objective of this thesis, and we will describe some cases only as examples of bottlenecks.

a. The CPU

The CPU is saturated when its utilization approaches 100 percent. A truly saturated CPU can be cured only by being replaced with a faster one. However, some further analysis may reveal different underlying causes for the saturation and suggest cures of a less dramatic nature.

One case in point occurs when the CPU becomes saturated due to paging activity. This occurs when total CPU utilization approaches 100 percent, problem state time declines, and paging rate climbs as the number of users increases. Such conditions prevail if, for some reason, the scheduler consistently underestimates working sets and thus maintains too high a MPL. Reducing the MPL will release some of the CPU time spent on paging, but whether or not the reduced MPL will be sufficiently high to maintain good throughput depends on the amount of storage available. Increasing storage capacity while retaining the same MPL would also decrease the paging rate and release some CPU time for productive use.

The presence of compute-bound jobs in the work load can result in very high CPU utilization. However, Some of the CPU time can be made available to other users, should they need it. Thus, if response time to the compute-bound tasks is not a primary concern, then we should not really consider the CPU to be saturated.

b. Main Storage

Main storage is (or at least the scheduler thinks that it is) saturated when the eligible list is almost never empty, [Ref.30]. Nevertheless, a saturated memory is not necessarily a performance bottleneck. If paging is moderate and the CPU is fully utilized, then main storage capacity is adequate and will have to be increased

only after a more powerful CPU is installed. If both paging and CPU utilization are light, then the scheduler is probably overestimating working sets and consequently maintaining too low a MPL.

If the paging rate is high, productive CPU utilization (% problem state time) is low, and if the MPL is high, then the scheduler may be at fault. This so-called trashing condition may be removed by inducing the scheduler to maintain a lower MPL.

Only if the MPL is low, paging is heavy, and productive CPU utilization (% problem state time) is low, is the saturated main storage a true bottleneck and in need of expansion.

c. Paging

There are three components which contribute to the total system wait time in the VM/370. The idle wait, when no high-speed I/O requests are outstanding. The page wait, when outstanding I/O requests are primarily for paging. The I/O wait, when outstanding I/O requests are not primarily for paging.

Typically, the idle-wait state predominates when the number of users is small. By the time saturation is reached, idle wait will have decreased to zero. Absence of idle wait in itself, however is no proof of saturation when the number of users is moderate, there may be enough work to keep either the CPU or an I/O path busy at any given time, but not enough to saturate either.

If the CPU is not the bottleneck, there will be a substantial amount of wait state even at saturation. This wait state may be due to poor overlapping of CPU and I/O activities caused by main storage being insufficient to

accommodate an adequate MPL. If this is not the case, however, and if page wait accounts for the major part of the residual wait time, then the paging subsystem is probably at fault.

Page wait may be experienced either because the paging rate is too high or because page transit time is too long. The first condition is caused by working sets being underestimated by the scheduler. The second occurs when either no high-speed paging devices are installed or their capacity has been exceeded.

d. I/O Activity

A bottleneck in the I/O subsystem reveals itself in a manner analogous to the paging subsystem. If there is enough main storage to maintain an adequate MPL, and yet a significant amount of I/O wait time remains at saturation, a deficient I/O subsystem is indicated. It may be simply that the work load is so I/O bound that no possible expansion of the I/O facilities will handle it. In this case, one might conclude that the CPU in use is too fast and a slower one would suffice. More typically, some rearrangement or expansion of the I/O subsystem will solve the problem.

Under extreme conditions, an I/O bottleneck may develop if the MPL is too high, rather than too low. If there is a concidence between very high MPL and high I/O wait time, one should amend the scheduler so as to restrict the MPL to a lower level.

D. DATA REQUIREMENTS

The foregoing analysis requires that performance and resource usage variables be measured over a certain time period, say, one or two weeks of operation.

For this particular experiment, The measurements have to be done using the tools available to a General User, which is the status of the author. This means that only priviledge class G commands are available. In Chapter V the VM/SP performance tools are described, most of which are not available to the General User.

The actual tools that are available for a General User and were used in the experiment are:

- The CP Indicate LOAD command (class G)
- The CP Indicate USER command (class G)

Other commands used to gather information are the following;

- CP QUERY VIRTUAL
- CP QUERY TERMINAL
- CP QUERY SET
- CP QUERY TIME
- CP QUERY USERS

Note: See Chapter V Section B for Indicate command formats.

The list of the performance and resource usage variables to be measured is next:

VARIABLE		TOOL
Response time	(seconds)	CP Q TIME
Expansion	(seconds)	CP IND LOAD
VIRT CPU	(seconds)	CP Q TIME
TOT CPU	(seconds)	CP Q TIME
CPU UTILIZATION	(%)	CP IND LOAD
APU UTILIZATION	(%)	CP IND LOAD
Storage	(%)	CP IND LOAD
Paging	(Pag/Sec)	CP IND LOAD

Residents	(Pages)	CP IND USER
Worksheet size	(Pages)	CP IND USER
SIO	(Count)	CP IND USER
Users	(Count)	CP Q USERS
Q1	(Count)	CP IN LOAD
Q2	(Count)	CP IN LOAD
ML	(Count)	(Q1 + Q2)

Note: to do a complete bottleneck analysis, some data required (% Problem state time, % Idle wait time, % Page wait time, % I/O wait time) is not obtainable with the use of priviledge class G (General User) commands. This represented a significant limitation for data gathering and analysis.

1. Variables Definition

a. CPU % (CP)

indicates the percentage of time that the main processor is running and is derived from the smoothed wait value maintained by the scheduler.

b. APU % (AP)

is the percentage of the time the attached processor is running.

c. Storage (ST)

is a measure of the usage of real storage. It is an approximation of the number of pages in storage for in-queue virtual machines to the number of pageable pages in the system, expressed as a percentage.

d. Expansion (EX)

is the scheduler contention value, and is the smoothed measure of the time it takes a virtual machine to

receive a given amount of processor time. It is a measure of the total delay in response time that a virtual machine experiences due to contention for both real storage and the processor. If there is no contention for either resource, EXPAN is maintained at a minimum value of 1 second.

e. Paging (PA)

is the average number of page I/O operations (page reads and writes) performed per second.

f. Residents (RES)

is the current number of the user's virtual storage pages resident in real storage at the time the command is issued.

g. Working Set (WS)

the system maintains an estimate of each user's storage requirements, this estimate is refered to as the user's projected working set.

h. SIO (SIO)

is the total number of nonpooled I/O requests issued by the user.

I. Q1 (Q1)

is the number of interactive tasks, i.e., those who require fast response while making only slight demands on system resources.

j. Q2 (Q2)

is the number of noninteractive tasks currently being serviced.

k. Multiprogramming Level (MPL)

is the number of tasks that may actually receive CPU time at any moment. It is equal to the number of tasks in Q1 and Q2 states.

1. Response time(RE)

Composed of the running time and the waiting time in the eligible list. This variable is going to be used as the performance variable of the system, in this analysis.

NOTE.- We'll use the symbols in parentheses to refer to the respective variables.

E. USER WORKLOAD CHARACTERIZATION

Two kinds of synthetic programs were chosen. Both algorithms are very simple, with the intention to avoid unnecessary complexity, being representative and sufficient for our purposes.

The first algorithm accesses a file of data and reads a value, executes additions the number of times that value indicates, and writes the result another file. In this algorithm access to the input and output files was included to control the execution.

The second algorithm accesses a file a number of times, reading some values each time and then writing them to another file.

These two algorithms have been written in FORTRANH, WATFIV, COBOL, and PASCAL.

Besides these algorithms, two script files are used for other types of tasks. Three file handling commands were tested: The ERASE, RENAME, and COPYFILE COMMANDS.

Considering the compilation and execution of the same

algorithm as different tasks, in the case of FORTRANH and COBOL, the tasks to be used as representatives of typical tasks are the following:

	Task Name	Task Type
1	FT 1 C	FortranH CPU task (compilaticn)
2	FT 1 E	FortranH CPU task (Execution)
3	FT 2 C	FortranH I/O task (compilaticn)
4	FT 2 E	FortranH I/O task (execution)
5	FT 3 C	FortranH
6	FT 3 E	FortranH
7	FT 4 C	FortranH
8	FT 4 E	FortranH
9	WT 1	Watfiv CPU task (Comp and Exe)
10	WT 2	Watfiv I/O task (comp and exe)
11	CT 1 C	Cobol CPU task (compilation)
12	CT 1 E	Cobol CPU task (execution)
13	CT 2 C	Cobol I/O task (compilation)
14	CT 2 E	Cobol I/O task (execution)
15	PT 1	Pascal CPU task (comp and exe)
16	PT 2	Pascal I/O task (comp and exe)
17	ST 1	Script task 1
18	ST 2	Script tast 2
19	RENAME	RENAME TASIK
20	COPYFILE	Copyfile task
21	ERASE	Erase task
22	MIXT	All the tasks abcve runing continously.

Note: tasks 5,6,7 and 8 correspcnd to tasks 1,2,3 and 4 executed using different compilaticn options.

F. TIMES AND NUMBER CF CBSERVATIONS

Four "shifts" were considered. The morning shift, which

is from 0800-1300; the afternoon shift from 1300-1800; the weekend shift from 0900 to 2100, saturdays and sundays, the 'fixed' shift which is from 1500 to 1530 hours, only on weekdays.

An observation (for our purposes) is defined as the procedure of data collection of the values of performance and resource usage variables for the 22 tasks indicated above.

Twenty five observations were made for each shift, which makes a total of one hundred observations. This means that each task will be executed 100 times, at different times of the day, and days of the week.

Given the number of tasks that had to be executed for each observation, an EXEC procedure was written in a way that allowed us to execute the 22 tasks, to gather performance information required for each one, continuosly, one after the other, and with only one command.

The creation of listing files, display of messages, and output generated by the different tasks, had to be avoided, when possible, in order to incur the least overhead, and to reduce considerably the output for each observation. All the information gathered by one observation is stored in a file created by spooling the console output and sending it to the printer.

G. DATA COLLECTION

The first observation was taken Nov 5, 1981 and the last one Nov 20, 1981. The observations were done at different times of the day, weekdays, and weekends, according to the number of observations required for each shift but without any particular pattern.

The different execs which were used are listed in Appendix A. The algorithms written in the four languages are described in Appendix B. The formats used to record the data and a sample of the actual data recorded for a particular task are shown in appendix C. A sample of one observation output is in Appendix D.

To take one observation as an example, the command used was:

DACO XXX 'SHIFT' YYY

where:

DACO stands for DATA collection and is the name of the Main Exec.
XXX is the number of the observations
'Shift' indicates Morning, afternoon, weekend, or fixed.
YYY is the number of observations by shifts.

After one observation was finished, the values of the performance and resource usage variables and additional information required for each task were taken from the 'observation output' and recorded in the forms of Appendix C.

Once the data for the 100 observations were written in the forms mentioned above, the raw data collection was finished.

H. THE USE OF "MINITAB"

"MINITAB" is a general purpose statistical computing system, designed especially for students and researchers primarily for moderate size data sets which can be stored in main memory. Minitab was found especially helpful for exploring data in the early phases of analysis, for plotting, and for regression analysis.

In order to use 'Minitab' an MTAB type file was created containing the data recorded for each task. After all these MTAB files were created, equal numbers of worksheets were created and saved, using MINITAB commands. At this point all the raw data gathered was ready to be processed.

Minitab has been available in the NPS Computer Center since the 31st August, 1981. For more information see the Minitab Reference Manual [Ref.29].

'Minitab' consists of a worksheet of rows and columns, in which data are stored, and a collection of about 150 commands, which operate on the data stored in the worksheet.

1. Data Reduction

For each task the correlation between every two performance and resource usage variables were calculated. Observing primarily those with higher correlation coefficients, some basic transformations such as log and square root were applied to the performance and resource usage variables.

Different plots depicting the relation between performance and resource usage variables were obtained and analyzed.

I. RESULTS

1. Correlation Coefficients

Samples of the tables of correlation coefficients obtained for each task are shown in Tables 1, 2.

```
MINITAB RELEASE 81.1 ***
COPYRIGHT - PENN STATE UNIV. 1981
*** NAVAL POSTGRADUATE SCHOOL (31 AUG 81) VER. 3
STORAGE AVAILABLE  40000
```

--

retr 'wft1c'

TABLE 1

--

corr c1-c15

	SH	OBS	RE	EX	CP	AP
OBS	-0.057					
RE	-0.434	-0.115				
EX	-0.554	0.138	0.593			
CP	-0.675	0.120	0.365	0.418		
AP	-0.752	-0.019	0.338	0.361	0.758	
ST	-0.598	0.242	0.537	0.808	0.728	0.535
PA	-0.572	0.276	0.440	0.757	0.598	0.527
RES	0.497	0.016	-0.449	-0.557	-0.540	-0.478
WS	0.504	-0.032	-0.458	-0.555	-0.557	-0.489
SIO	-0.102	-0.101	-0.047	0.077	0.196	0.244
US	-0.644	0.124	0.499	0.727	0.654	0.661
Q1	-0.595	0.115	0.669	0.895	0.590	0.442
Q2	-0.657	0.099	0.669	0.814	0.746	0.641
MPL	-0.649	0.109	0.689	0.874	0.698	0.571

	ST	PA	RES	WS	SIO	US
PA	0.804					
RES	-0.618	-0.688				
WS	-0.638	-0.696	0.987			
SIO	0.187	0.129	-0.132	-0.098		
US	0.710	0.893	-0.640	-0.649	0.109	
Q1	0.898	0.820	-0.601	-0.617	0.065	0.773
Q2	0.926	0.764	-0.643	-0.651	0.178	0.749
MPL	0.941	0.811	-0.643	-0.655	0.132	0.782

	Q1	Q2
Q2	0.885	
MPL	0.962	0.978

--

retr 'wmix'

TABLE 2

--

corr c1-c15

	SH	OBS	RE	EX	CP	AP
OBS	-0.057					
RE	-0.514	-0.027				
EX	-0.554	0.138	0.718			
CP	-0.675	0.120	0.533	0.418		
AP	-0.752	-0.019	0.446	0.361	0.758	
ST	-0.598	0.242	0.700	0.808	0.728	0.535
PA	-0.572	0.276	0.617	0.757	0.598	0.527
RES	0.497	0.016	-0.573	-0.557	-0.540	-0.478
WS	0.504	-0.032	-0.578	-0.555	-0.557	-0.489
SIO	-0.102	-0.101	0.067	0.077	0.196	0.244
US	-0.644	0.124	0.630	0.727	0.654	0.661
Q1	-0.595	0.115	0.779	0.895	0.590	0.442
Q2	-0.657	0.099	0.801	0.814	0.746	0.641
MPL	-0.649	0.109	0.815	0.874	0.698	0.571

	ST	PA	RES	WS	SIO	US
PA	0.804					
RES	-0.618	-0.688				
WS	-0.638	-0.696	0.987			
SIO	0.187	0.129	-0.132	-0.098		
US	0.710	0.893	-0.640	-0.649	0.109	
Q1	0.898	0.820	-0.601	-0.617	0.065	0.773
Q2	0.926	0.764	-0.643	-0.651	0.178	0.749
MPL	0.941	0.811	-0.643	-0.655	0.132	0.782

	Q1	Q2
Q2	0.885	
MPL	0.962	0.978

The results obtained are the following: for Tasks 1 to 16 and 22 the higher correlation coefficients are the following:

BETWEEN	AVERAGE
RE and EX	.59
RE and ST	.59
RE and US	.53
RE and MPL	.70
EX and MPL	.85
CP and Q2	.77
ST and MPL	.93
PA and US	.89
CP and US	.65
A^ and VS	.67

For the Tasks 17, 18, 19 and 21 the response time obtained was always less or equal to 1 second, independently of the load on the system. The time information gathered by the CP QUERY TIME commands have precision of the order of seconds. Thus, no further analysis can be done on these tasks in terms of response time and its relation with other performance variables, using these commands as the source of data for Response time. The sizes of the two script files used by Tasks 17 and 18 are in relation of 1 to 10. Even so, it was not possible to note differences in the response times for these tasks. Obviously, the RENAME and ERASE tasks are very trivial and the tools used to record response time does not allow us to establish a relation between the response time and the load on the system.

For the Task 20, 'COPYFILE' the only moderately high correlation coeffiecient, considering Response time, was: RE VS SIO, .524 . Some other results were consistent with

those obtained for Tasks 1 to 16 and 22. That is, the compilations, executions and the Copyfile tasks. They are :

EX	and	MPL	.857
CP	and	Q2	.781
AP	and	US	.683
ST	and	MPL	.936
PA	and	US	.893
WS	and	RES	.987
CP	and	US	.659

When the plots are made, they may immediately reveal some clear trends, but sometimes, however, the trends will be masked by random fluctuations in the data. This is most likely to occur in those portions of the curves where relatively few observations are available, typically in the upper and lower ranges of user values.

Other reasons for these fluctuations are that in this experiment the system workload variations have not been minimized; as a General user, there is no tool available to measure these variations.

In the worst case, there are fluctuations in system workload that vary from I/O oriented to CPU oriented.

2. Relation Between MPL and CPU Utilization

As shown in Figure 6, the trend corresponds to the situation shown in Figure 3.

The saturation point is affected by the nature of the work-load of the system. We can assume that for the top points of Figure 6 the work load is CP oriented demanding more CP utilization. We can see that with an MPL of 5 the CP utilization reaches approximately 96%.

In contrast the bottom points are considered to be I/O workload oriented, which demands more I/O activity than CPU utilization. Obviously, for an I/O oriented work load, the CPU utilization will not rise up to a saturation point. An example of this is the observation that with an MPL of 13 the CPU utilization is approximately only 55%.

```
--
retr 'wft1c'
--
plot 'cp' 'mpl'
```

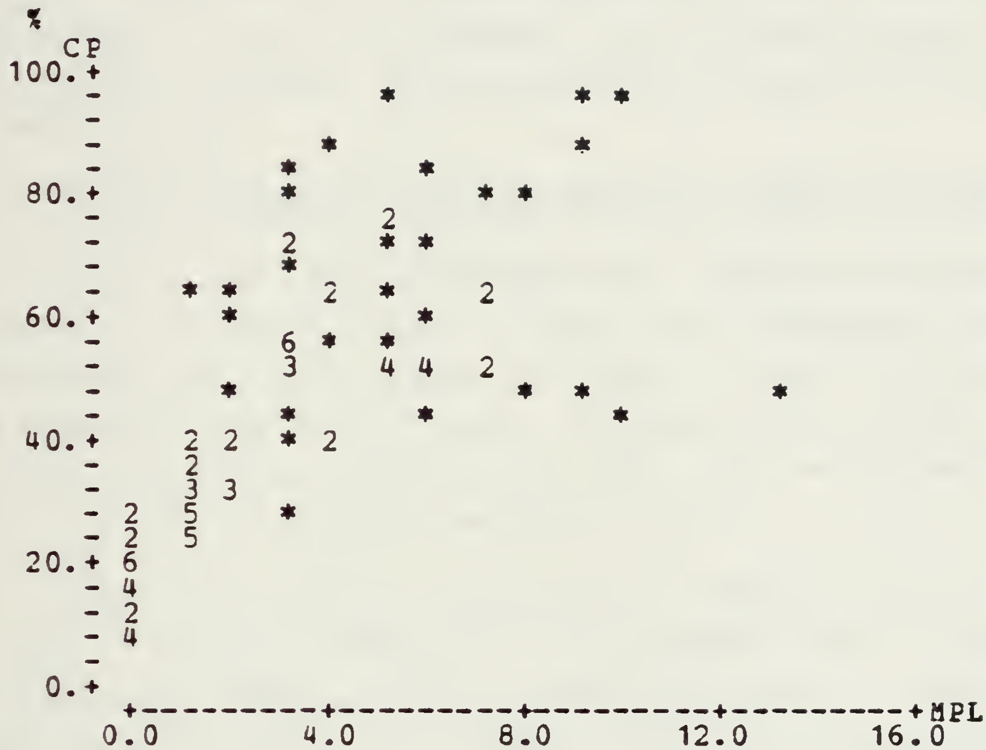

FIG. 6

RELATION BETWEEN CPU UTILIZATION AND MPL

3. Relation Between MPL and AP utilization

The plot of Figure 7 shows that the AP is saturated with an MPL of about 3. If we compare the plots of Figures 6 and 7 we can apreciate that the AP is more utilized than the CPU, which may indicate that load leveling is not effective. Furthermore, it could indicate that the workload of the system is not I/O oriented, as one would expect for this particular system, given that the AP does not process I/O instructions.

4. Relation Between Response Time and number of Users

The plot of Figure 8 shows the essential effect of saturation on response time. Below the saturation point, approximately 86, the system is able to serve additional users without severely impacting response time. With more than about 86 users, additional users can be accomodated only at the cost of reduced service to all.

Again this analysis applies to a typical task, i.e., one whose various resource requirements are roughly proportional to those of the overall work-load. Radically different work loads, e.g., compute bound or I/O bound ones, may affect the response profile for a particular task when related to the number of users.

The plot of Figure 8 corresponds to the FortranH Task 1, and is representative of all the compilation and execution tasks even for different languages.

If we observe the trend of Figure 8 it is very similar to the typical curve shown in Figure 4, and was obtained by real measurements on the system.

--
plot 'AP' 'MPL'

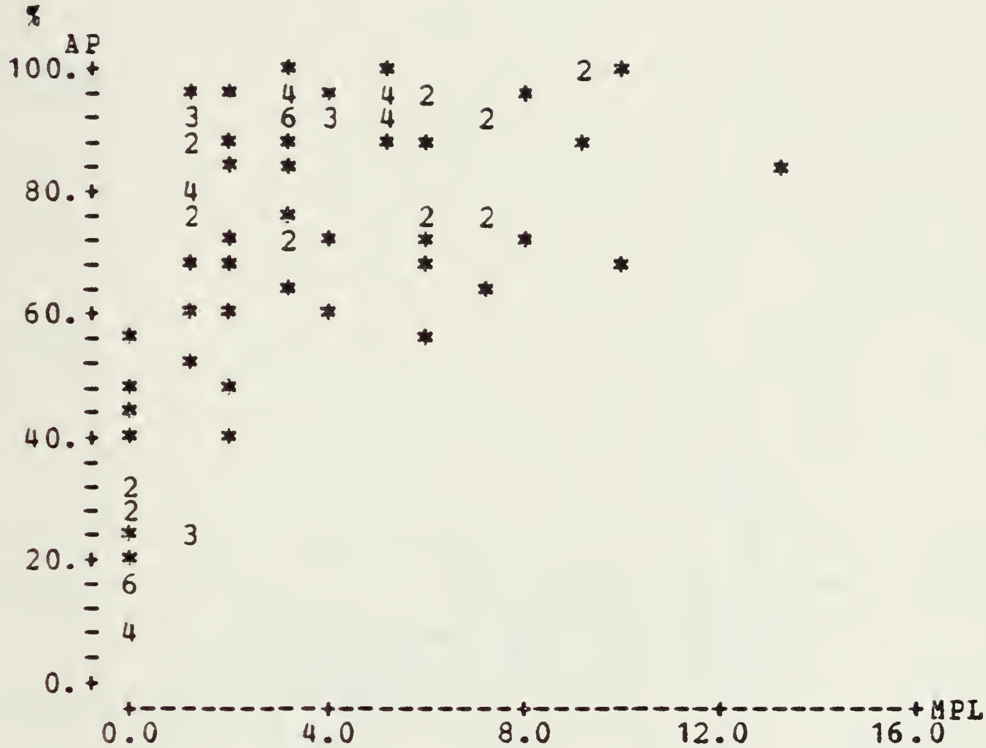

FIG.7

RELATION BETWEEN APU UTILIZATION AND MPL

```
--
--
plot 're' 0 50 'us' 0 110
```

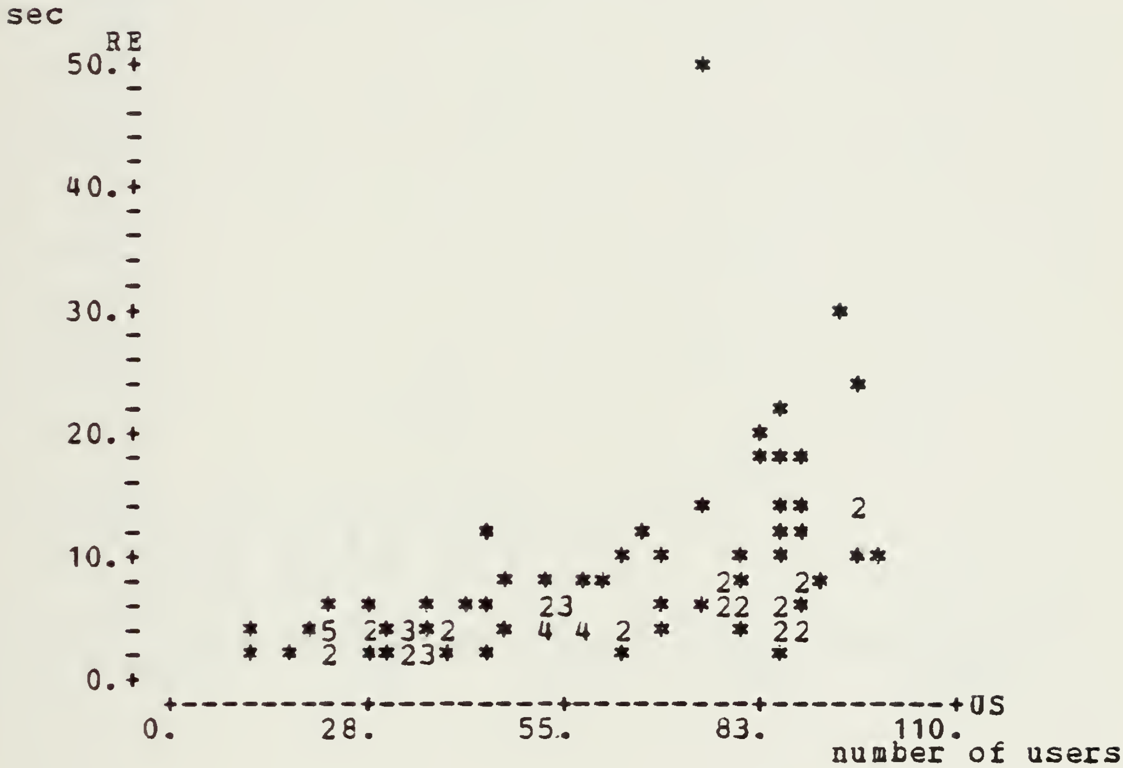

FIG.8

RELATION BETWEEN RESPONSE TIME AND NUMBER OF USERS

plot 'RE' 'MPL'

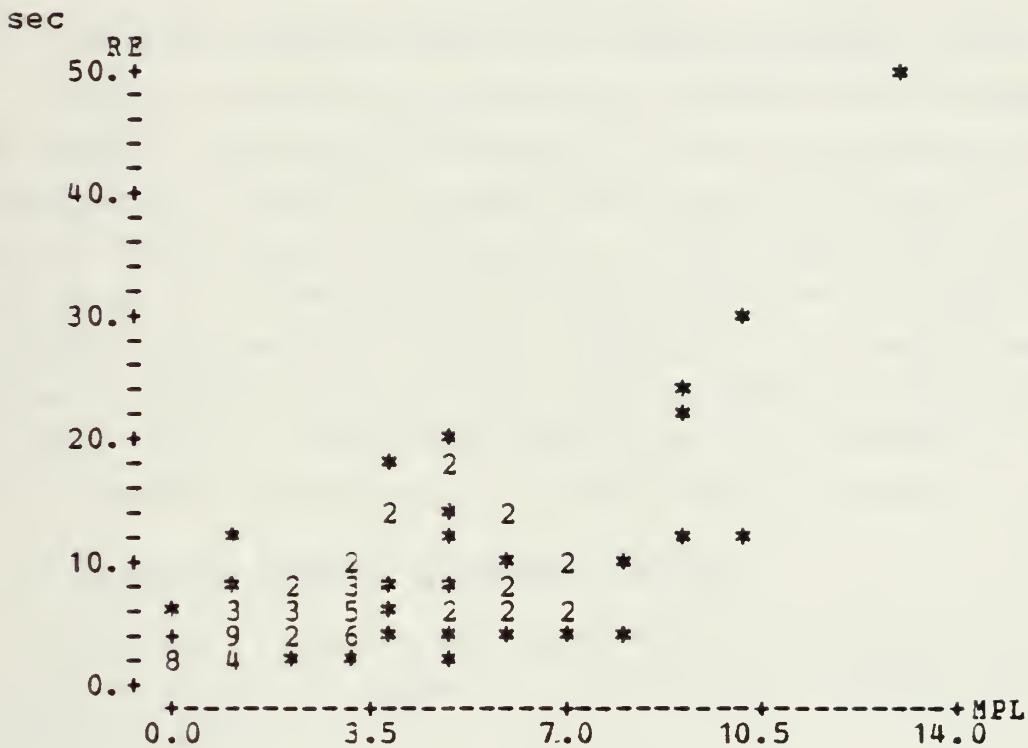

FIG. 9

RELATION BETWEEN RESPONSE TIME AND MPL

5. Relation Between Response Time and MPL

The plot of Figure 9 shows that the response time rises with a very slight increment in MPL until MPL reaches the value of about 9. After that, the effect of increase in MPL produces a very high increase in response time.

6. Relation Between Users and Multiprogramming Level

Figure 10 shows the relation between MPL and Users. If we observe the trend and project it, we can estimate that the saturation point is around 105 and that even if the number of users exceed this number, The higher MPL will be around 14 or 15. Obviously, if this occurs, more tasks will be pending and waiting in the eligible list before being serviced, and the response time will increase rapidly for a particular task. Note that with an MPL of around 9, the number of users varies from 65 to 95 (for this data).

7. Relation Between Expansion and MPL

The MPL is somewhat correlated with Expansion and the plot of Figure 11 allows us to estimate expansion based on values of MPL.

--
plot 'MPL' 'US'

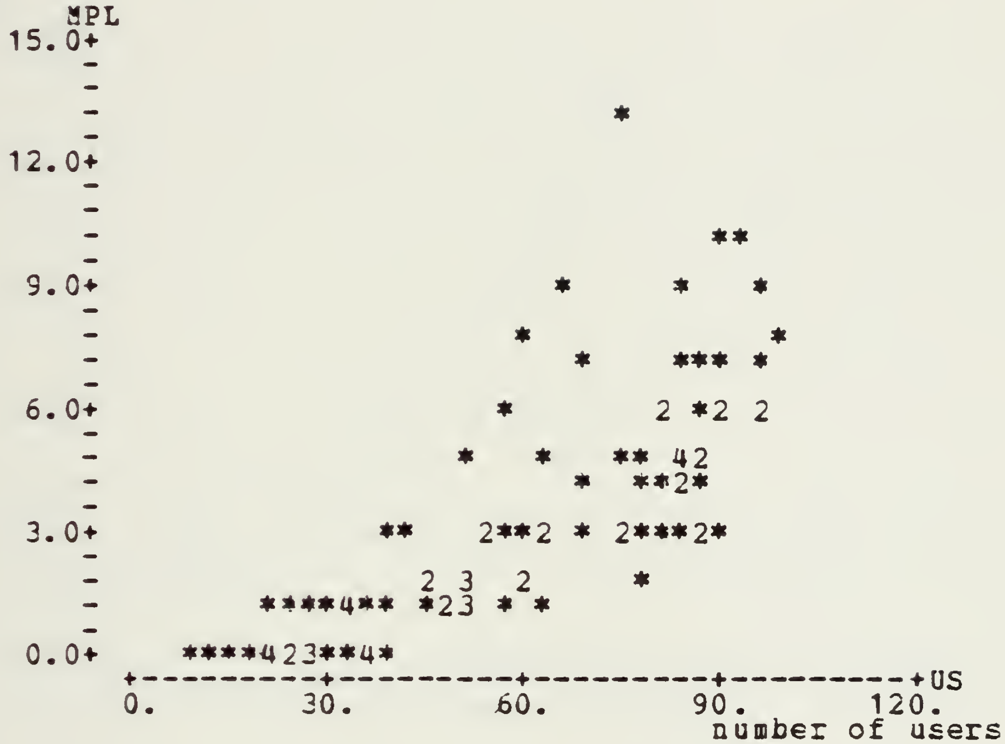

FIG.10

RELATION BETWEEN NUMBER OF USERS AND MPL

```
--
plot 'EX' 'MPL'


sec
     EX
   10.0+
       -
       -                          *
       -
       -
    8.0+
       -
       -                        *         *
       -
       -
    6.0+                     *
       -
       -           *    2  3
       -
       -
    4.0+         *    * 5  2 *  * *
       -
       -    * *  2 2  8 *       *
       -
       -
    2.0+    8 6  + 3  *
       -
       - +  9 *  *
       -
       -
    0.0+
         +---------+---------+---------+---------+MPL
        0.0       4.0       8.0      12.0      16.0
```

FIG. 11

RELATION BETWEEN EXPANSION AND MPL

J. REGRESSION ANALYSIS

A major difficulty in the evaluation of computer systems has been the inability to provide a valid quantification of the relationship between performance and work load [Ref.31]. the problem of the presence of a large number of variables is compounded when the interactions of these variables must be taken into account.

A regression equation can serve as a predictor of performance and resource usage within the range of the variables which were used to estimate the regression coefficients [Ref.32]. A disadvantage results from treating computer functions as black boxes, where the regression variables are black box inputs and outputs. This macro approach fails to deal with the internal structure of computer functions. The characteristics of the Internal structures may be important determinants of computer performance. Also, it is possible to have a very good fit between dependent and Independent variables without a cause and effect reason for the relationship. The goodness of the fit may mislead one to believe that the mathematical relationship implies a physical relationship.

1. Simple Regression Equations

The correlation analysis showed some high correlation coefficients, and we considered those for the regression equations.

Regression equations were obtained for all those relations that showed high correlation, with emphasis on the Performance variable, Response time, and those independent variables which are the major contributors for its prediction.

A summary of the regression equations which were

developed appears in Tables 3 to 6 for four different tasks.

We can see that the variable with highest contribution to the variation of Response time is the Multiprogramming level. The mean of its contribution for the tasks used in the experiment is: 54 %.

TABLE 3

TASK 'FTIC'

DEP VAR	INDEP VAR	r	r^2	SS x10-2	MS=SS/DF X10-2
RE	EX	.59	.351	15.70	15.70
RE	ST	.53	.288	12.87	12.87
RE	US	.49	.249	11.12	11.12
RE	MPL	.68	.474	21.22	21.22
EX	MPL	.87	.764	1.84	1.84
CP	Q2	.74	.556	263.50	273.50
ST	MPL	.94	.885	58.62	58.62
PA	US	.89	.798	193.50	193.50
WS	RES	.98	.975	79.22	79.22
AP	US	.66	.437	354.70	354.70

TABLE 4

TASK 'MIX'

DEP VAR	IND VAR	r	R^2	SS X10-4	MS=SS/DF X10-4
RE	EX	.71	.516	31.99	31.99
RE	ST	.69	.489	30.34	30.34
RE	US	.63	.397	24.61	24.61
RE	MPL	.81	.664	41.14	41.14
				X10-2	
EX	MPL	.87	.764	1.84	1.84
CP	Q2	.74	.556	273.50	273.50
ST	MPL	.94	.885	58.62	58.62
PA	US	.89	.798	193.50	193.50
WS	RES	.98	.975	79.22	79.22
AP	US	.66	.437	354.70	354.70

TABLE 5

TASK 'CT1C'

DEP VAR	IND VAR	r	R^2	SS X10-2	MS=SS/DF X10-2
RE	EX	.57	.326	18.36	18.36
RE	ST	.62	.387	21.77	21.77
RE	US	.53	.284	15.99	15.99
RE	MPL	.69	.484	27.23	27.23
EX	MPL	.85	.728	2.03	2.03
CP	Q2	.77	.600	304.20	304.20
ST	MPL	.93	.874	67.52	67.52
PA	US	.89	.809	199.60	199.60
WS	RES	.14	.02	2.18	2.18
AP	US	.67	.451	366.80	366.80

TABLE 6

TASK CT2E

DEV VAR	IND VAR	r	R^2	SS X10-2	MS=SS/DF X10-2
RE	EX	.58	.338	4.28	4.28
RE	ST	.63	.401	5.07	5.07
RE	US	.55	.306	3.87	3.87
RE	MPL	.74	.559	7.08	7.08
EX	MPL	.85	.724	2.00	2.00
CP	Q2	.77	.602	304.20	304.20
ST	MPL	.93	.876	66.14	66.14
PA	US	.89	.805	195.06	195.06
WS	RES	.39	.159	28.97	28.97
AP	US	.68	.467	384.83	84.80

2. Multiple Regression Equations

This approach was taken to investigate the possibility of improving the forecast by using more than one predictor variable.

A summary of the regression equations which were developed appears in Tables 7 to 10 for four different tasks.

It is easy to observe in these tables that the major contributions to the variation of response time come from the number of Users, Multiprogramming Level and Storage, where MPL, alone, is almost as good as the three variables combined.

Calculating the mean for the different tasks, we can show the contribution of the three variables above, using a pie chart:

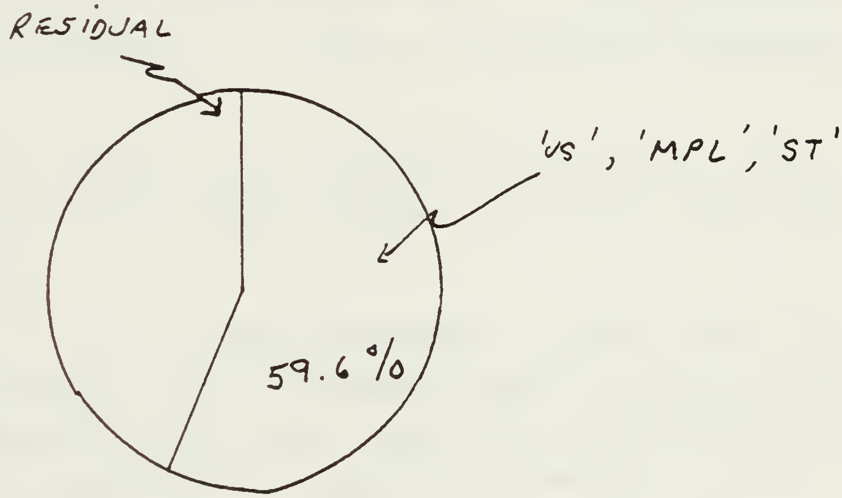

FIGURE. 12

MULTIPLE CONTRIBUTION TO THE VARIATION IN 'RE'

The contribution of each variable is shown below:

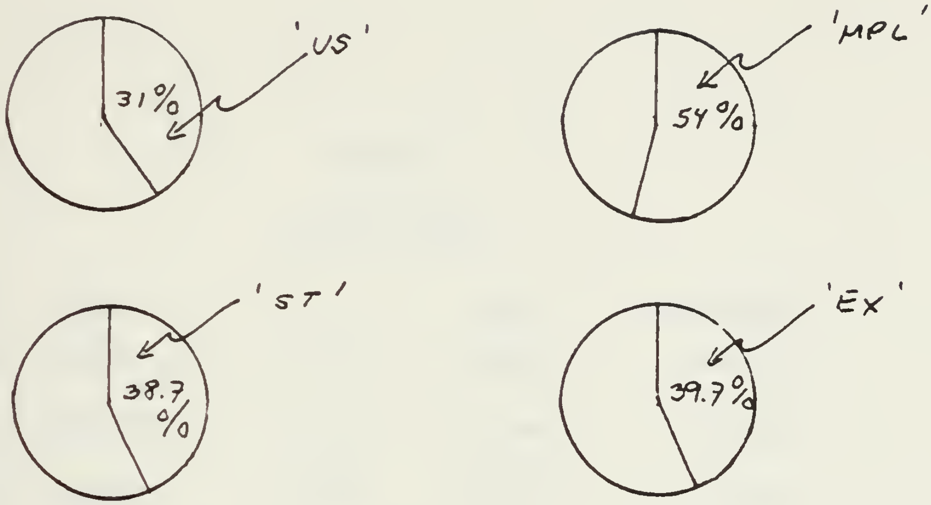

FIGURE. 13

SIMPLE CONTRIBUTION TO THE VARIATION IN 'RE'

If we include 'Exp' as the 4th variable, the contribution only rises from 59.6 to 59.7. This indicates that 'Exp' is very dependent on other independent variables such as storage.

Now, if we compare the contribution of 'MPL', 'US', 'ST' together with that of 'MPL' alone, we can see that the difference is 6%.

Therefore, the Multiprogramming level can be considered the variable with the highest contribution to the variation in response time, thus the best load indicator found with the data gathered for this experiment.

The correlation between 'US' and 'MPL' is .78. Thus, since the number of users is easy to obtain, this could also be considered as a predictor (not accurate in all the cases) of response time.

TABLE 7

TASK 'FT1C'

DEP VAR	IND VAR	R^2	SS X10-2	MS=SS/DF X10-2
RE	US,ML	.479	21.40	10.70
RE	US,ML ST	.593	26.52	8.84
RE	US,ML ST,EX	.594	26.57	6.64
RE	US,ML ST,EX,PA	.602	26.91	5.38
RE	US,ML,ST EX,PA,CP	.614	27.45	4.57
RE	US,ML,ST EX,PA,CP AP	.615	27.50	3.92

TABLE 8

TASK'MIX'

DEP VAR	IND VAR	R^2	SS X10-4	MS=SS/DF X10-4
RE	US,ML	.664	41.14	20.57
RE	US,ML ST	.704	43.61	14.53
RE	US,ML ST,EX	.704	43.61	10.90
RE	US,ML ST,EX,PA	.704	43.62	8.72
RE	US,ML,ST EX,PA,CP	.704	43.65	7.27
RE	US,ML,ST EX,PA,CP AP	.705	43.70	6.24

TABLE 9

TASK 'CT1C'

DEP VAR	IND VAR	R^2	SS X10-2	MS=SS/DF X10-2
RE	US,ML	.485	27.28	13.64
RE	US,ML ST	.491	27.66	9.22
RE	US,ML ST,EX	.493	27.75	6.93
RE	US,ML ST,EX,PA	.504	28.34	5.66
RE	US,ML,ST EX,PA,CP	.508	28.57	4.76
RE	US,ML,ST EX,PA,CP AP	.510	28.69	4.09

TABLE 10

TASK 'CT2E'

DEP VAR	IND VAR	R^2	SS X10-2	MS=SS/DF X10-2
RE	US,ML	.563	7.13	3.56
RE	US,ML ST	.60	7.61	2.53
RE	US,ML ST,EX	.607	7.69	1.92
RE	US,ML ST,EX,PA	.607	7.69	1.53
RE	US, ML,ST EX,PA,CP	.607	7.70	1.28
RE	US,ML,ST EX,PA,CP AP	.607	7.70	1.10

APPENDIX A

DACO EXEC

```
**********************************************************
THIS IS THE MAIN EXEC USED TO RUN EACH OBSERVATION
**********************************************************


 &TRACE
 GLOBAL TXTLIB COBLIBVS
 ACC 19E P
 *OBSERVATION    103
 *   FIXED       025
 *   MORNING     025
 *   AFTERNOON   025
 *   WEEKEND     025
 EXEC BLK5
 CP SPOOL CON TO * START NOTERM CONT CLASS A
 &TYPE
 &TYPE
 &TRACE
 &A = &LITERAL OF OBSERVATION
 &B = &LITERAL OF  *****
 &TYPE &A &1 &B &2 &3
 EXEC BLK5
 CP Q T
 &TYPE
 &TYPE
 &TYPE
 &TYPE SYSTEM DESCRIPTION
 EXEC BLK5
 CP Q SET
 &TYPE
 CP Q TERMINAL
 &TYPE
 CP Q VIRTUAL
 &TYPE
 CP Q PROCESS
 &TYPE
 CP Q SPMODE
 &TYPE
 CP Q TIME
 &TYPE
 CP Q USERS
 &TYPE
 &TYPE
 EXEC BLK20
 EXEC BLK5
 ****** FORTRANH TASK 1
 EXEC FORTAS1
 &TYPE
 EXEC BLK5
 ****** FORTRANH TASK 2
 EXEC FORTAS2
 &TYPE
 EXEC BLK10
 ****** FORTRANH TASK 3
 EXEC FORTAS3
 &TYPE
 EXEC BLK5
 ****** FORTRANH TASK 4
 EXEC FORTAS4
```

```
&TYPE
EXEC BLK5
****** WATFIV TASK 1
EXEC WATTAS1
&TYPE
EXEC BLK10
EXEC BLK20
****** WATFIV TASK 2
EXEC WATTAS2
&TYPE
EXEC BLK20
EXEC BLK20
****** COBOL TASK 1
EXEC COBTAS1
&TYPE
EXEC BLK10
****** COBOL TASK 2
EXEC COBTAS2
&TYPE
EXEC BLK20
****** PASCAL TASK 1
EXEC PASTAS1
&TYPE
EXEC BLK5
EXEC BLK20
****** PASCAL TASK 2
EXEC PASTAS2
&TYPE
EXEC BLK10
EXEC BLK20
****** SCRIPT TASK 1
EXEC SCRTAS1
&TYPE
EXEC BLK20
EXEC BLK20
******* SCRIPT TASK 2
EXEC SCRTAS2
&TYPE
&TYPE
&TYPE
EXEC BLK10
****** MISCELLANEOUS TASKS
EXEC MISTAS
CP SPOOL CON STOP CLOSE
CP SPOOL CON OFF
READ CONSOLE FILE
Q TIME
PRINT CONSOLE FILE (LINECOUN 77
```

```
**************************
THIS IS THE EXEC FORTAS1
**************************

&TRACE
&PRINT FORTRANH TASK 1
EXEC DACO1
FORTHX FORTAS1 (NOPRINT
EXEC DACO1
GLOBAL TXTLIB FORTMOD2 MOD2EEH
FILEDEF 01 DISK FT1D DATA A1 (RECFM FB LRECL 80 PERM
FILEDEF 02 DISK FT1O DATA A1 (RECFM FB LRECL 80 PERM
&PRINT
&PRINT
LOAD FORTAS1 (START
EXEC DACO1

**************************
THIS IS THE EXEC WATTAS1
**************************

&TRACE
&PRINT WATFIV TASK 1
EXEC DACO1
FILEDEF 06 CLEAR
FILEDEF 01 DISK WT1D WATFIV A1 (RECFM FB LRECL 80 PERM
&TYPE
WATFIV WATTAS1 WT1D
EXEC DACO1


**************************
THIS IS THE EXEC COBTAS1
**************************


&TRACE
&PRINT COBOL TASK 1
&TYPE
&TYPE
GLOBAL TXTLIB COBLIBVS
ACC 19E P
FILEDEF CARDIN DISK CT1D DATA (PERM
FILEDEF CARDOU DISK CT1O DATA (PERM
EXEC DACO1
COBOL COBTAS1
EXEC DACO1
LOAD COBTAS1 (START
EXEC DACO1


**************************
THIS IS THE EXEC PASTAS1
**************************

&TRACE
&PRINT PASCAL TASK 1
EXEC DACO1
PW PASTAS1
EXEC DACO1
```

APPENDIX B

PROGRAMS CODE

****** THIS IS THE FIRST FORTRANH ALGORITHM ******

```
C      THIS IS THE FIRST FORTRANH TASK
       SUM=0
       READ(01,*)J
C      READING A VALUE FROM AN INPUT FILE
       DO 50 I=1,J
C         EXECUTING AN ADDITION J TIMES
          SUM=SUM+100
  50   CONTINUE
       WRITE(02,*)SUM
C      WRITING A VALUE TO AN OUTPUT FILE
       STOP
       END
```

****** THIS IS THE SECOND FORTRANH ALGORITHM ******

```
C      THIS IS THE FORTRANH TASK 2
       SUM = 0
       DO 50 I=1,20
C      READING FROM AN INPUT FILE AND WRITING TO AN OUTPUT FILE
          READ(01,*)A,B,C,D,E
          WRITE(02,*)A,B,C,D,E
          SUM = SUM +A+B+C+D+E
  50   CONTINUE
       AVER=SUM/100
       WRITE(02,*)AVER
C      TO CHECK ALL THE DATA WAS READ CORRECTLY
       STOP
       END
           $JOB
```

****** THIS IS THE FIRST WATFIV ALGORITHM ******

```
      C      THIS IS THE FIRST WATFIV TASK
             SUM=0
             READ(01,*)J
             DO 50 I=1,J
                SUM=SUM+100
        50   CONTINUE
             WRITE(06,*) SUM
             STOP
             END
      $ENTRY
      $JOB
```

****** THIS IS THE SECOND WATFIV ALGORITHM ******

```
C       THIS IS THE WATFIV TASK 2
        SUM=0
        DO 50 I=1,20
           READ(01,*) A,B,C,D,E
           WRITE(06,*) A,B,C
           WRITE(06,*) D,E
           SUM=SUM+A+B+C+D+E
   50   CONTINUE
        AVER=SUM/100
        WRITE(06,*) AVER
        STOP
        END
$ENTRY
```

****** THIS IS THE FIRST COBOL ALGORITHM ******

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. CCETAS1.
       AUTHOR. WOML.
           SKIP1
      *    THIS IS THE FIRST COBOL TASK
           SKIP1
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. IBM-3033.
       OBJECT-COMPUTER. IBM-3033.
           SKIP1
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT IN-DATA ASSIGN TO DA-S-CARDIN.
           SELECT OUT-DATA ASSIGN TO UR-S-CARDOU.

       DATA DIVISION.
       FILE SECTION.
       FD  IN-DATA LABEL RECORDS ARE STANDARD.
       01  NUM1.
           05 NUM                          PIC 9(4).
           05 FILLER                       PIC X(76).
       FD  OUT-DATA LABEL RECORDS ARE OMITTED.
       01  TOTAL                          PICTURE 999999.
       WORKING-STORAGE SECTION.
       77  SUMTOT                           PICTURE 999999.
       77  REPS                             PICTURE 99999.
       PROCEDURE DIVISION.
           SKIP1
       CONTROL-ROUTINE.
           SKIP1
           PERFORM INITIALIZE.
           PERFORM SUMATION UNTIL REPS IS EQUAL TO NUM.
           PERFORM CLEANUP.
           STOP RUN.
       INITIALIZE.
           OPEN INPUT IN-DATA.
           OPEN OUTPUT OUT-DATA.
           MOVE ZERO TO SUMTOT, REPS, TOTAL.
           READ IN-DATA AT END STOP RUN.
       SUMATION.
           ADD 100 TO SUMTOT.
           ADD 1 TO REPS.
       CLEANUP.
           WRITE TOTAL FROM SUMTOT.
```

```
          CLOSE IN-DATA.
          CLOSE OUT-DATA.

****** THIS IS THE SECOND COBOL ALGORITHM ******

       IDENTIFICATION DIVISION.
       PROGRAM-ID. CCETAS2.
       AUTHOR. WOML.
           SKIP1
      *    THIS IS THE SECOND COBOL TASK
           SKIP1
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. IBM-3033.
       OBJECT-COMPUTER. IBM-3033.
           SKIP1
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT IN-DATA ASSIGN TO DA-S-CARDIN.
           SELECT OUT-DATA ASSIGN TO UR-S-CARDOU.

       DATA DIVISION.
       FILE SECTION.
       FD  IN-DATA LABEL RECORDS ARE STANDARD.
       01  INUMS.
           05 NUM1                            PIC 9.
           05 NUM2                            PIC 9.
           05 NUM3                            PIC 9.
           05 NUM4                            PIC 9.
           05 NUM5                            PIC 9.
           05 FILLER                          PIC X(75).
       FD  OUT-DATA LABEL RECORDS ARE OMITTED.
       01  ONUMS.
           05 NUM1                            PIC 9.
           05 NUM2                            PIC 9.
           05 NUM3                            PIC 9.
           05 NUM4                            PIC 9.
           05 NUM5                            PIC 9.
       01  AVER                               PIC 9.
       WORKING-STORAGE SECTION.
       77  SUMTOT                              PIC 999.
       77  COUN                                PIC 99 VALUE IS 20.
       77  REPS                                PIC 99.
       PROCEDURE DIVISION.
           SKIP1
       CONTROL-ROUTINE.
           SKIP1
           PERFORM INITIALIZE.
           PERFORM READ-WRITE UNTIL REPS IS EQUAL TO COUN.
           PERFORM CLEANUP.
           STOP RUN.
       INITIALIZE.
           OPEN INPUT IN-DATA.
           OPEN OUTPUT OUT-DATA.
           MOVE ZERO TO SUMTOT, REPS, AVER.
       READ-WRITE.
           READ IN-DATA.

           WRITE ONUMS FROM INUMS.
           ADD NUM1 OF INUMS TO SUMTOT.
           ADD NUM2 OF INUMS TO SUMTOT.
           ADD NUM3 OF INUMS TO SUMTOT.
           ADD NUM4 OF INUMS TO SUMTOT.
           ADD NUM5 OF INUMS TO SUMTOT.
           ADD 1 TO REPS.
       CLEANUP.
```

```
        COMPUTE SUMTOT = SUMTOT / 100.
        WRITE AVER FROM SUMTCT.
        CLOSE IN-DATA.
        CLOSE OUT-DATA.
```

****** THIS IS THE FIRST PASCAL ALGORITHM ******

```
(* THIS IS THE FIRST PASCAL TASK *)
PROGRAM PASTAS1 (INPUT,OUTPUT);
VAR SUM : INTEGER;
    NUM : INTEGER;
    REP : INTEGER;
      X : FILE OF CHAR;
      Y : FILE OF CHAR;
BEGIN
    (* OPENING INPUT AND OUTPUT FILES *)
    RESET(X,'PT1D DATA A');
    REWRITE (Y,'PT1C DATA A');
    WRITELN;WRITELN;
    SUM:=0;NUM:=0;REP:=0;
    READ(X,NUM);
    WHILE REP <= NUM DO
       BEGIN SUM :=SUM+100;REP:=REP+1 END;
    WRITELN(Y,SUM);
END.
```

****** THIS IS THE SECOND PASCAL ALGORITHM ******

```
(* THIS IS THE SECOND PASCAL TASK *)
PROGRAM PASTAS2 (INPUT,OUTPUT);
CONST  M=100;
VAR SUM : INTEGER;
    NUM : INTEGER;
    TIM : INTEGER;
    AVER: INTEGER;
    A,B,C,D,E: INTEGER;
      X : FILE OF CHAR;
      Y : FILE OF CHAR;
BEGIN
   (* OPENING INPUT AND OUTPUT FILES *)
    RESET(X,'PT2D DATA A');
    REWRITE (Y,'PT2C DATA A');
    WRITELN;WRITELN;
    SUM := 0;AVER := 0;TIM := 0;
    REPEAT  READ(X,A);
            WRITE(Y,A);
            SUM := SUM+A;
            TIM := TIM+1;
            WRITELN(Y,TIM)
    UNTIL TIM=100;
    AVER := SUM DIV 100;
    WRITE(Y,AVER);WRITELN(Y,SUM)
END.
```

APPENDIX C

SAMPLE DATA

```
***********************************************************************
THIS IS SAMPLE DATA OF THE FIRST FORTRANH TASK - COMPILATION
***********************************************************************

    MINITAB RELEASE 81.1
    *** COPYRIGHT - PENN STATE UNIV. 1981
    *** NAVAL POSTGRADUATE SCHOOL (31 AUG 81) VER. 3
    STORAGE AVAILABLE  40000

    --
    retr 'wft1c'
    --
    outputwidth 60
    --
    print c1-c15
    COLUMN        SH          OBS          RE          EX
    COUNT        100          100         100         100
    ROW
      1           6.           1.         30.          4.
      2           1.           2.         22.          4.
      3           1.           3.         49.          7.
      4          13.           4.          8.          3.
      5          13.           5.          4.          2.
      6           6.           6.          4.          2.
      7           6.           7.          5.          3.
      8           1.           8.          2.          2.
      9          13.           9.          9.          2.
     10          13.          10.          7.          1.
     11           6.          11.          5.          4.
     12           6.          12.          8.          3.
     13           1.          13.          4.          2.
     14          13.          14.          5.          2.
     15          13.          15.          4.          2.
     16          13.          16.          5.          4.
     17           6.          17.          4.          1.
     18           1.          18.          7.          2.
     19           1.          19.          3.          1.
     20          13.          20.          5.          1.
     21          13.          21.          3.          2.
     22          13.          22.          3.          2.
     23          13.          23.          4.          2.
     24           6.          24.         17.          3.
     25           6.          25.          4.          3.
     26          26.          26.          3.          1.
     27          26.          27.          2.          1.
     28          26.          28.          2.          1.
     29          26.          29.          2.          1.
     30          26.          30.          2.          1.
     31          26.          31.          3.          1.
     32          26.          32.          2.          1.
     33          26.          33.          3.          1.
     34          26.          34.          3.          1.
     35          26.          35.          3.          1.
     36          26.          36.          3.          1.
     37          26.          37.          2.          1.
     38          26.          38.          3.          1.
```

39	26.	39.	2.	1.
40	13.	40.	4.	2.
41	13.	41.	3.	2.
42	6.	42.	9.	5.
43	6.	43.	14.	5.
44	6.	44.	14.	5.
45	6.	45.	9.	6.
46	1.	46.	12.	9.
47	1.	47.	12.	3.
48	1.	48.	6.	3.
49	1.	49.	20.	3.
50	1.	50.	8.	3.
51	1.	51.	5.	2.
52	1.	52.	11.	1.
53	1.	53.	2.	1.
54	13.	54.	9.	2.
55	13.	55.	5.	2.
56	13.	56.	5.	2.
57	6.	57.	10.	4.
58	6.	58.	5.	4.
59	1.	59.	6.	1.
60	1.	60.	17.	2.
61	1.	61.	13.	3.
62	1.	62.	7.	2.
63	1.	63.	5.	2.
64	1.	64.	6.	2.
65	1.	65.	3.	2.
66	26.	66.	2.	3.
67	26.	67.	6.	1.
68	26.	68.	4.	1.
69	26.	69.	3.	1.
70	26.	70.	4.	1.
71	26.	71.	2.	1.
72	26.	72.	2.	1.
73	26.	73.	3.	1.
74	26.	74.	3.	1.
75	26.	75.	5.	1.
76	26.	76.	3.	2.
77	13.	77.	2.	2.
78	13.	78.	8.	2.
79	13.	79.	4.	2.
80	13.	80.	13.	2.
81	13.	81.	3.	2.
82	13.	82.	2.	2.
83	13.	83.	3.	2.
84	13.	84.	8.	2.
85	1.	85.	14.	5.
86	1.	86.	6.	3.
87	1.	87.	24.	7.
88	1.	88.	3.	1.
89	6.	89.	17.	4.
90	6.	90.	7.	4.
91	6.	91.	5.	3.
92	6.	92.	2.	3.
93	6.	93.	11.	3.
94	6.	94.	4.	4.
95	6.	95.	4.	3.
96	6.	96.	8.	4.
97	6.	97.	5.	5.
98	6.	98.	4.	5.
99	6.	99.	6.	4.
100	6.	100.	9.	4.

COLUMN COUNT ROW	CP 100	AP 100	ST 100	PA 100
1	95.	100.	24.	45.
2	95.	100.	23.	37.
3	48.	83.	28.	14.
4	40.	92.	12.	30.
5	87.	92.	14.	11.
6	66.	92.	9.	13.
7	57.	73.	12.	19.
8	30.	81.	5.	3.
9	56.	94.	10.	15.
10	61.	69.	9.	5.
11	64.	93.	22.	17.
12	50.	92.	11.	18.
13	31.	51.	4.	1.
14	39.	60.	4.	4.
15	49.	86.	6.	13.
16	40.	77.	9.	12.
17	81.	94.	8.	17.
18	32.	70.	6.	14.
19	62.	94.	6.	6.
20	23.	90.	3.	10.
21	27.	79.	4.	13.
22	37.	69.	6.	9.
23	35.	75.	6.	9.
24	51.	90.	11.	35.
25	50.	91.	12.	34.
26	21.	43.	1.	0.
27	18.	32.	1.	0.
28	17.	17.	2.	1.
29	17.	15.	3.	1.
30	28.	81.	6.	1.
31	26.	60.	6.	1.
32	7.	6.	0.	0.
33	8.	6.	1.	0.
34	8.	8.	1.	0.
35	9.	9.	2.	0.
36	16.	57.	1.	0.
37	15.	47.	2.	0.
38	12.	33.	1.	0.
39	11.	26.	1.	0.
40	65.	95.	9.	8.
41	56.	91.	9.	18.
42	50.	76.	17.	31.
43	45.	66.	16.	32.
44	50.	76.	19.	32.
45	49.	70.	28.	53.
46	42.	66.	25.	38.
47	50.	86.	13.	50.
48	57.	90.	13.	28.
49	72.	93.	14.	30.
50	52.	94.	11.	22.
51	57.	93.	11.	14.
52	38.	91.	7.	7.
53	25.	80.	3.	2.
54	29.	91.	6.	14.
55	56.	93.	9.	19.
56	32.	49.	7.	23.
57	50.	74.	17.	35.
58	52.	72.	19.	32.
59	25.	77.	2.	1.
60	62.	94.	9.	20.
61	77.	97.	15.	26.
62	82.	98.	12.	12.
63	94.	99.	19.	11.
64	71.	73.	12.	6.
65	57.	62.	10.	5.

66	25.	24.	1.	1.
67	24.	22.	2.	1.
68	25.	38.	1.	1.
69	26.	29.	3.	1.
70	26.	20.	2.	1.
71	20.	16.	2.	1.
72	19.	14.	3.	1.
73	19.	14.	4.	1.
74	20.	16.	5.	1.
75	25.	22.	6.	1.
76	27.	23.	7.	1.
77	31.	85.	5.	6.
78	29.	87.	5.	5.
79	32.	88.	4.	6.
80	64.	92.	18.	33.
81	52.	94.	7.	30.
82	45.	95.	7.	28.
83	50.	83.	8.	25.
84	73.	89.	8.	21.
85	39.	58.	14.	26.
86	39.	38.	4.	4.
87	46.	89.	28.	49.
88	39.	93.	4.	1.
89	64.	94.	15.	34.
90	60.	94.	18.	37.
91	57.	72.	12.	41.
92	75.	96.	20.	33.
93	89.	98.	31.	30.
94	78.	96.	32.	28.
95	83.	97.	28.	25.
96	50.	55.	25.	53.
97	53.	63.	24.	55.
98	64.	77.	23.	47.
99	70.	86.	20.	40.
100	81.	92.	23.	35.

COLUMN COUNT ROW	RES 100	WS 100	SIO 100	US 100
1	23.	23.	76.	93.
2	22.	21.	81.	84.
3	54.	53.	77.	74.
4	31.	30.	85.	77.
5	54.	53.	81.	70.
6	50.	49.	83.	64.
7	53.	52.	80.	80.
8	57.	56.	79.	38.
9	54.	53.	80.	68.
10	55.	54.	82.	51.
11	39.	38.	83.	70.
12	22.	21.	80.	76.
13	55.	54.	82.	31.
14	55.	54.	81.	52.
15	55.	54.	83.	59.
16	55.	54.	84.	55.
17	53.	52.	81.	59.
18	55.	54.	78.	61.
19	56.	56.	78.	48.
20	55.	54.	78.	56.
21	52.	52.	81.	64.
22	55.	53.	78.	52.
23	58.	57.	76.	51.
24	47.	46.	77.	87.
25	43.	42.	79.	86.
26	55.	54.	76.	21.
27	58.	58.	79.	21.
28	55.	54.	76.	37.
29	58.	58.	79.	35.
30	55.	54.	76.	34.
31	55.	53.	79.	34.
32	55.	53.	72.	23.
33	58.	58.	77.	23.
34	58.	57.	84.	21.
35	58.	58.	81.	21.
36	55.	54.	81.	19.
37	58.	57.	78.	16.
38	58.	58.	78.	11.
39	58.	58.	80.	10.
40	47.	46.	83.	51.
41	51.	50.	80.	81.
42	53.	52.	80.	96.
43	52.	51.	78.	96.
44	52.	51.	80.	97.
45	34.	33.	76.	99.
46	33.	32.	80.	89.
47	31.	31.	82.	85.
48	42.	41.	80.	85.
49	45.	44.	76.	83.
50	54.	53.	80.	79.
51	35.	35.	83.	55.
52	51.	50.	80.	45.
53	56.	55.	78.	34.
54	55.	54.	82.	63.
55	46.	45.	80.	74.
56	43.	41.	78.	78.
57	41.	36.	58.	81.
58	43.	43.	80.	80.
59	58.	57.	80.	27.
60	51.	47.	78.	83.
61	46.	43.	81.	74.
62	48.	46.	76.	57.
63	48.	47.	78.	52.
64	49.	48.	79.	41.
65	54.	53.	79.	39.

66	55.	53.	82.	36.
67	58.	57.	76.	37.
68	56.	55.	76.	38.
69	55.	54.	76.	35.
70	58.	57.	76.	34.
71	58.	57.	76.	30.
72	58.	57.	76.	28.
73	58.	57.	76.	27.
74	58.	57.	80.	27.
75	58.	58.	80.	23.
76	58.	58.	76.	21.
77	55.	54.	76.	45.
78	55.	55.	76.	47.
79	53.	51.	77.	51.
80	41.	34.	78.	88.
81	55.	52.	72.	89.
82	55.	54.	79.	86.
83	52.	51.	83.	84.
84	51.	50.	79.	87.
85	41.	40.	82.	84.
86	55.	52.	58.	45.
87	40.	35.	81.	97.
88	55.	53.	81.	33.
89	43.	42.	78.	85.
90	29.	28.	81.	91.
91	41.	37.	81.	78.
92	55.	54.	82.	62.
93	39.	28.	83.	65.
94	53.	52.	82.	59.
95	48.	47.	83.	57.
96	42.	41.	80.	89.
97	50.	49.	79.	89.
98	50.	49.	83.	87.
99	52.	51.	83.	86.
100	42.	41.	80.	84.

COLUMN	Q1	Q2	ML
COUNT	100	100	100
ROW			
1	4.	6.	10.
2	3.	6.	9.
3	5.	8.	13.
4	1.	3.	4.
5	1.	3.	4.
6	1.	2.	3.
7	2.	2.	4.
8	0.	1.	1.
9	1.	2.	3.
10	1.	1.	2.
11	2.	5.	7.
12	1.	2.	3.
13	0.	1.	1.
14	1.	1.	2.
15	1.	1.	2.
16	1.	2.	3.
17	1.	2.	3.
18	1.	1.	2.
19	0.	1.	1.
20	0.	1.	1.
21	0.	1.	1.
22	0.	1.	1.
23	0.	1.	1.
24	2.	3.	5.
25	2.	3.	5.
26	0.	0.	0.
27	0.	0.	0.
28	0.	0.	0.
29	0.	0.	0.
30	0.	1.	1.
31	0.	1.	1.
32	0.	0.	0.
33	0.	0.	0.
34	0.	0.	0.
35	0.	0.	0.
36	0.	0.	0.
37	0.	0.	0.
38	0.	0.	0.
39	0.	0.	0.
40	0.	2.	2.
41	1.	2.	3.
42	3.	4.	7.
43	3.	3.	6.
44	3.	3.	6.
45	4.	4.	8.
46	5.	5.	10.
47	2.	3.	5.
48	2.	3.	5.
49	2.	3.	5.
50	2.	3.	5.
51	1.	2.	3.
52	0.	1.	1.
53	0.	1.	1.
54	1.	2.	3.
55	1.	2.	3.
56	1.	1.	2.
57	3.	3.	6.
58	3.	3.	6.
59	0.	1.	1.
60	2.	2.	4.
61	2.	3.	5.
62	1.	2.	3.
63	1.	4.	5.
64	1.	2.	3.
65	1.	2.	3.

66	0.	1.	1.
67	0.	0.	0.
68	0.	0.	0.
69	0.	0.	0.
70	0.	0.	0.
71	0.	0.	0.
72	0.	0.	0.
73	0.	0.	0.
74	0.	0.	0.
75	0.	1.	1.
76	0.	1.	1.
77	0.	2.	2.
78	0.	1.	1.
79	0.	1.	1.
80	1.	3.	4.
81	1.	2.	3.
82	1.	2.	3.
83	1.	2.	3.
84	1.	2.	3.
85	2.	2.	4.
86	1.	1.	2.
87	4.	5.	9.
88	0.	1.	1.
89	2.	3.	5.
90	2.	4.	6.
91	1.	2.	3.
92	2.	3.	5.
93	4.	5.	9.
94	3.	5.	8.
95	2.	4.	6.
96	3.	3.	6.
97	3.	4.	7.
98	3.	4.	7.
99	2.	4.	6.
100	2.	5.	7.

--
stop

APPENDIX D

SAMPLE OF OBSERVATION OUTPUT

```
*******************************************
THIS IS THE SAMPLE OF AN OBSERVATION OUTPUT
*******************************************


OBSERVATION 000  ***** SAMPLE 0Q1

TIME IS 11:15:26 PST WEDNESDAY 12/09/81
CONNECT= 01:02:12 VIRTCPU= 000:06.64 TOTCPU= 000:15.27


SYSTEM DESCRIPTION


MSG ON , WNG ON , EMSG TEXT, ACNT ON , RUN OFF
LINEDIT ON , TIMER ON  , ISAM OFF, ECMODE OFF
ASSIST ON SVC   NOTMR, PAGEX OFF, AUTOPOLL OFF
IMSG ON , SMSG OFF, AFFINITY NONE   , NOTRAN OFF
VMSAVE OFF, 370E OFF
STBYPASS OFF    , STMULTI OFF 0Q/000

LINEND  # , LINEDEL   , CHARDEL  @ , ESCAPE  " , TABCHAR ON
LINESIZE 080, ATTN OFF, APL OFF, TEXT OFF, MODE VM
HILIGHT OFF, CONMODE 3215, BREAKIN   IMMED , BRKKEY PA1
SCRNSAVE OFF

STORAGE = 01024K
CHANNELS = SEL
CONS 009 ON GRAF 47B  NOTERM START
     009 CL A    CONT NOHOLD COPY 001   READY FORM STANDARD
```

```
     009 TO  2807P     DIST MARMANIL  FLASHC 000
     009 FLASH        CHAR       MDFY       FCB
RDR  00C CL *  NOCONT NOHOLD   EOF       READY
PUN  00D CL A  NOCONT NOHOLD COPY 001   READY FORM STANDARD
     00D FOR 2807P    DIST MARMANIL
PRT  00E CL A  NOCONT NOHOLD COPY 001   READY FORM STANDARD
     00E FOR 2807P    DIST WOML       FLASHC 000
     00E FLASH        CHAR       MDFY       FCB
DASD 120 3350 VMRES2 R/O      6 CYL
DASD 121 3350 VMRES2 R/O      6 CYL
DASD 190 3350 VMRES1 R/O     45 CYL
DASD 191 3330 MS0083 R/W      8 CYL
DASD 19E 3350 VMRES1 R/O     65 CYL

PROCESSOR 00 ONLINE, PROCESSOR 01 ONLINE, SYSTEM IN AP MODE

SPMODE OFF

TIME IS 11:15:26 PST WEDNESDAY 12/09/81
CONNECT= 01:02:12 VIRTCPU= 000:06.65 TOTCPU= 000:15.31

079 USERS, 000 DIALED, 000 NET



FORTRANH TASK 1



CP IND LOAD
CPU  - 054%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0050 WS-0047 READS=001044 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:15 SIO=001551 RDR-000451
```

```
PRT-002453 PCH-000000

CP Q USERS
079 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:27 PST WEDNESDAY 12/09/81
CONNECT= 01:02:12 VIRTCPU= 000:06.66 TOTCPU= 000:15.36


FORTRAN H EXTENDED COMPILER ENTERED


*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******



CP IND LOAD
CPU   - 053%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0160 WS-0047 READS=001045 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:16 SIO=001639 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
079 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:29 PST WEDNESDAY 12/09/81
CONNECT= 01:02:16 VIRTCPU= 000:06.72 TOTCPU= 000:15.57
```

```
EXECUTION BEGINS...


CP IND LOAD
CPU    - 053%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0066 WS-0047 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:16 SIO=001759 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:36 PST WEDNESDAY 12/09/81
CONNECT= 01:02:23 VIRTCPU= 000:06.80 TOTCPU= 000:15.80


FORTRANH TASK 2


CP IND LOAD
CPU    - 053%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0067 WS-0047 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:16 SIO=001764 RDR-000451
PRT-002453 PCH-000000
```

```
CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:37 PST WEDNESDAY 12/09/81
CONNECT= 01:02:23 VIRTCPU= 000:06.81 TOTCPU= 000:15.83


FORTRAN H EXTENDED COMPILER ENTERED


*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******



CP IND LOAD
CPU   - 053%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0166 WS-0169 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:16 SIO=001849 RDR-000451
PRT-002453 PCH-00000C

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:39 PST WEDNESDAY 12/09/81
CONNECT= 01:02:25 VIRTCPU= 000:06.88 TOTCPU= 000:16.04


EXECUTION BEGINS...
```

```
CP IND LOAD
CPU    - 053%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0066 WS-0169 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:16 SIO=001977 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:42 PST WEDNESDAY 12/09/81
CONNECT= 01:02:28 VIRTCPU= 000:06.96 TOTCPU= 000:16.27
```

FORTRANH TASK 3

```
CP IND LOAD
CPU    - 053%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0067 WS-0169 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:16 SIO=001982 RDR-000451
PRT-002453 PCH-000000
```

```
CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:42 PST WEDNESDAY 12/09/81
CONNECT= 01:02:28 VIRTCPU= 000:06.98 TOTCPU= 000:16.32


FORTRAN H EXTENDED COMPILER ENTERED


*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******



CP IND LOAD
CPU   - 053%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0166 WS-0169 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:17 SIO=002061 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:44 PST WEDNESDAY 12/09/81
CONNECT= 01:02:30 VIRTCPU= 000:07.04 TOTCPU= 000:16.52


EXECUTION BEGINS...
```

```
CP IND LOAD
CPU    - 052%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0066 WS-0169 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:17 SIO=002181 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:47 PST WEDNESDAY 12/09/81
CONNECT= 01:02:33 VIRTCPU= 000:07.11 TOTCPU= 000:16.73
```

FORTRANH TASK 4

```
CP IND LOAD
CPU    - 052%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0067 WS-0169 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:17 SIO=002186 RDR-000451
PRT-002453 PCH-000000
```

```
CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:47 PST WEDNESDAY 12/09/81
CONNECT= 01:02:33 VIRTCPU= 000:07.12 TOTCPU= 000:16.76


FORTRAN H EXTENDED COMPILER ENTERED


*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******



CP IND LOAD
CPU   - 052%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0166 WS-0132 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:17 SIO=002265 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:50 PST WEDNESDAY 12/09/81
CONNECT= 01:02:37 VIRTCPU= 000:07.18 TOTCPU= 000:16.97



EXECUTION BEGINS...
```

```
CP IND LOAD
CPU   - 052%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0066 WS-0132 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:17 SIO=002387 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:54 PST WEDNESDAY 12/09/81
CONNECT= 01:02:40 VIRTCPU= 000:07.25 TOTCPU= 000:17.18



WATFIV TASK 1



CP IND LOAD
CPU   - 052%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0067 WS-0132 READS=001051 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:17 SIO=002392 RDR-000451
PRT-002453 PCH-000000
```

```
CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:54 PST WEDNESDAY 12/09/81
CONNECT= 01:02:40 VIRTCPU= 000:07.26 TOTCPU= 000:17.21



CP IND LOAD
CPU   - 052%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0066 WS-0132 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:17 SIO=002434 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:55 PST WEDNESDAY 12/09/81
CONNECT= 01:02:42 VIRTCPU= 000:07.30 TOTCPU= 000:17.38


WATFIV TASK 2


CP IND LOAD
CPU   - 052%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
```

```
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0066 WS-0132 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:17 SIO=002438 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:55 PST WEDNESDAY 12/09/81
CONNECT= 01:02:42 VIRTCPU= 000:07.32 TOTCPU= 000:17.45

CP IND LOAD
CPU    - 052%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0066 WS-0136 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:18 SIO=002491 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:57 PST WEDNESDAY 12/09/81
CONNECT= 01:02:43 VIRTCPU= 000:07.38 TOTCPU= 000:17.64
```

COBOL TASK 1

```
'19E' REPLACES ' P (19E) '
P (19E) R/O
19E ALSC = Y-DISK


CP IND LOAD
CPU   - 052%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0066 WS-0136 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:07 TTIME=000:18 SIO=002574 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:15:58 PST WEDNESDAY 12/09/81
CONNECT= 01:02:45 VIRTCPU= 000:07.45 TOTCPU= 000:17.81


REL2.3 OS/VS COBOL IN PROGRESS


CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
```

```
PAGES: RES-0079 WS-0136 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:18 SIO=002728 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:03 PST WEDNESDAY 12/09/81
CONNECT= 01:02:49 VIRTCPU= 000:07.65 TOTCPU= 000:18.19

EXECUTION BEGINS...



CP IND LOAD
CPU    - 051%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0067 WS-0136 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:18 SIO=002790 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:05 PST WEDNESDAY 12/09/81
CONNECT= 01:02:51 VIRTCPU= 000:07.70 TOTCPU= 000:18.33



COBOL TASK 2
```

```
CP IND LOAD
CPU    - 051%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0067 WS-0136 READS=0Q1057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:18 SIO=0Q2795 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:05 PST WEDNESDAY 12/09/81
CONNECT= 01:02:51 VIRTCPU= 000:07.72 TOTCPU= 000:18.37



REL2.3 OS/VS COBOL IN PROGRESS



CP IND LOAD
CPU    - 051%  APU    - 094%  Q1-Q1  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0079 WS-0098 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:19 SIO=002974 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
```

```
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:10 PST WEDNESDAY 12/09/81
CONNECT= 01:02:56 VIRTCPU= 000:07.95 TOTCPU= 000:18.80

EXECUTION BEGINS...



CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%

EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0067 WS-0098 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:19 SIO=003045 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:12 PST WEDNESDAY 12/09/81
CONNECT= 01:02:58 VIRTCPU= 000:07.99 TOTCPU= 000:18.92



PASCAL TASK 1



CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%
```

```
CP IND USER
PAGES: RES-0067 WS-0098 READS=0Q1057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:19 SIO=003051 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:12 PST WEDNESDAY 12/09/81
CONNECT= 01:02:59 VIRTCPU= 000:08.01 TOTCPU= 000:18.97

Execution begins...



...execution ends
File 'PASTAS1': 18 lines; no diagnostics
410 bytes of object code generated
3013 statements executed
10928 bytes of memory requested during compilation
5792 bytes returned before execution
10704 bytes requested during execution



CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0099 WS-0098 READS=001057 WRITES=000874
ME -0000  PH -0051
VTIME=000:08 TTIME=000:19 SIO=003102 RDR-000451
```

```
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:14 PST WEDNESDAY 12/09/81
CONNECT= 01:03:00 VIRTCPU= 000:08.18 TOTCPU= 000:19.25



PASCAL TASK 2



CP IND LOAD
CPU    - 051%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0099 WS-0098 READS=0Q1057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:19 SIO=0Q3106 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:14 PST WEDNESDAY 12/09/81
CONNECT= 01:03:01 VIRTCPU= 000:08.20 TOTCPU= 000:19.30

Execution begins...



...execution ends
File 'PASTAS2': 25 lines; no diagnostics
```

```
578 bytes of object code generated
510 statements executed
10928 bytes of memory requested during compilation
5792 bytes returned before execution
10704 bytes requested during execution



CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0100 WS-0098 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:20 SIO=003171 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:16 PST WEDNESDAY 12/09/81
CONNECT= 01:03:02 VIRTCPU= 000:08.36 TOTCPU= 000:19.56



SCRIPT TASK 1



CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
```

```
PAGES: RES-0100 WS-0098 READS=0Q1057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:20 SIO=0Q3175 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:16 PST WEDNESDAY 12/09/81
CONNECT= 01:03:02 VIRTCPU= 000:08.38 TOTCPU= 000:19.63



CP IND LOAD
CPU    - 051%  APU    - 094%  Q1-Q1  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0101 WS-0105 READS=0Q1057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:20 SIO=0Q3201 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
073 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:17 PST WEDNESDAY 12/09/81
CONNECT= 01:03:03 VIRTCPU= 000:08.39 TOTCPU= 000:19.72



SCRIPT TASK 2
```

```
CP IND LOAD
CPU    - 051%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0101 WS-0105 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:20 SIO=003207 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:17 PST WEDNESDAY 12/09/81
CONNECT= 01:03:03 VIRTCPU= 000:08.41 TOTCPU= 000:19.79


CP IND LOAD
CPU    - 051%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0101 WS-0105 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:20 SIO=003228 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:17 PST WEDNESDAY 12/09/81
CONNECT= 01:03:04 VIRTCPU= 000:08.43 TOTCPU= 000:19.87
```

MISCELLANEOUS TASKS

ERASE TASK

```
CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0101 WS-0105 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:20 SIO=003233 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:17 PST WEDNESDAY 12/09/81
CONNECT= 01:03:04 VIRTCPU= 000:08.44 TOTCPU= 000:19.92


CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0101 WS-0105 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:20 SIO=003240 RDR-000451
PRT-002453 PCH-000000
```

```
CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:18 PST WEDNESDAY 12/09/81
CONNECT= 01:03:04 VIRTCPU= 000:08.45 TOTCPU= 000:19.95



COPYFILE TASK



CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0101 WS-0105 READS=001057 WRITES=000874
MH -0000  PH -0051
VTIME=000:08 TTIME=000:20 SIO=003243 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:18 PST WEDNESDAY 12/09/81
CONNECT= 01:03:04 VIRTCPU= 000:08.47 TOTCPU= 000:20.01


CP IND LOAD
CPU   - 051%  APU   - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
```

```
PAGES: RES-0068 WS-0105 READS=001057 WRITES=000874
MH -0000  FH -0051
VTIME=000:09 TTIME=000:20 SIO=003405 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:21 PST WEDNESDAY 12/09/81
CONNECT= 01:03:08 VIRTCPU= 000:08.69 TOTCPU= 000:20.38



RENAME TASK



CP IND LOAD
CPU    - 051%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0068 WS-0105 READS=001057 WRITES=000874
MH -0000  FH -0051
VTIME=000:09 TTIME=000:20 SIO=003408 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:21 PST WEDNESDAY 12/09/81
CONNECT= 01:03:08 VIRTCPU= 000:08.71 TOTCPU= 000:20.45

CP IND LOAD
CPU    - 051%  APU    - 094%  Q1-01  Q2-02  STORAGE-008%
```

```
EXPAN-001
PAGING-018/SEC  STEAL-000%  LOAD-000%

CP IND USER
PAGES: RES-0068 WS-0105 READS=001057 WRITES=000874
MH -0000  FH -0051
VTIME=000:09 TTIME=000:20 SIO=003413 RDR-000451
PRT-002453 PCH-000000

CP Q USERS
078 USERS, 000 DIALED, 000 NET

CP Q TIME
TIME IS 11:16:21 PST WEDNESDAY 12/09/81
CONNECT= 01:03:08 VIRTCPU= 000:08.72 TOTCPU= 000:20.47
```

REFERENCES

1. Naval Postgraduate School Report 55Ss750J1, Analysis of Computer Performance in Multiprogrammed Processing , by Norman F. Schneidewind, may 1975.

2. Datapro Research Corporation, The EDP Buyer's bible, VOL 1, p. 70C-491-08a, McGraw - Hill Company, 1980.

3. Ibid., p. 70C-491-08a.

4. Ibid., P. 70C-491-08j.

5. IBM Corporation, A Guide to the IBM 4341 Processor , GC20-1877-0, p. 14.

6. Ibid., PP. 2-4.

7. Ibid., PP. 46-47.

8. IBM Virtual/Machine/System Product: Introduction; pp. 7-8.

9. IBM Corporation VM/SP System Programmer's Guide , sc19-6203-0, p 94.1.

10. Ibid., p 89.

11. Ibid., p 90.

12. ibid., p 96.

13. IBM Virtual/Machine/System Product Introduction; pp 14-15.

14. IBM Corporation VM/SP System Programmer's Guide , sc19-6203-0, pp 103-104.

15. IBM Corporation A Guide to the 3033 Processor Complex, AP Complex, and MP Complex , ga22-7060-0, p 108.

16. Ibid., p. 121.

17. IBM Corporation IBM Virtual Machine/System product: System Programmer's Guide ,sc19-6203-0, P. 116.

18. Performance Measurement tools for VM/370 , by P.H. Callaway., IBM Systems Journal Vol. 14 Number 2 ,1975.

19. IBM Corporation IBM Virtual Machine/System Product: System Programmer's Guide ,sc19-6203-0, p. 117.

20. Ibid., pp. 118-119.

21. Ibid., pp. 127-136.

22. Ibid., p. 142.

23. Ibid., pp. 279-295.

24. Y. Bard, Performance criteria and measurement for a time-sharing system , IBM Systems Journal 10, No.3, 1971.

25. Y. Bard. Experimental Evaluation of system performance, IBM systems Journal 12, No.3, 1973.

26. Y. Bard Performance Analysis of Virtual memory time-sharing systems , IBM Systems Journal 14, No 4, p. 368, 1975.

27. Ibid.,p. 370

28. Ibid.,p. 371

29. The Pennsylvania State University, MINITAB Reference Manual , by Thomas A. Ryan Jr.,Brian L. Joiner, Barbara F. Ryan, March 1981.

30. Y. Bard Performance Analysis of Virtual memory time-sHAring system , IBM System Journal 14, No 4, P. 373, 1975.

31. W. Freiberger (ed.), Statistical Computer performance Evaluation, Academic Press, New York, Quantititive Methods for evaluating Computer System Performance: A review and Proposals , pp. 3-24, 1972.

32. Naval Postgraduate School Report 55Ss750J1, Analysis of Computer Performance in Multiprogrammed Processing , by Norman F. Schneidewind, may 1975.

INITIAL DISTRIBUTION LIST

		Copies
1.	Defense Technichal Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 72 Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
4.	Computer Center Director, Code 0141 W. R. Church Computer Center Naval Postgraduate School Monterey, California 93940	1
5.	Mr. Steve Oxman Bldg. 132T U.S. Navy Trident CCSMA Newport, RI 02840	1
6.	Mr. Bill Ferreira Bldg. 12T U.S. Navy Trident CCSMA Newport, RI 02840	1
7.	Professor Norman Schneidewind, Code 54Ss Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
8.	Professor Uno Kodres, Code 52Kr Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
9.	LT. Waldo Marmanillo Lazo, Peruvian Navy Central de Procesamiento de Datos Ministerio de Marina Av. La Marina s/n, Maranga Lima - PERU	1
10.	Direccion de Instruccion Ministerio de Marina Av. Salaverry s/n, San Felipe Lima - PERU	1
11.	LT. Javier De La Cuba B. Central de Procesamiento de Datos Ministerio de Marina Av. La Marina s/n Lima - Peru	1

Thesis
L346
c.1

195396

Lazo
Performance analysis
of computer installations
virtual machine/370
(VM/370).