GUIDE TO VAX SPM

Order Number: AA–G139B–TE

June 1989

The *Guide to VAX SPM* presents conceptual and tutorial information for users who wish to collect and report data for system performance analysis and historical reporting. VAX SPM (Software Performance Monitor) provides tools for both on-line and manual performance evaluation, historical reporting, and billing for system use on VAX/VMS systems.

Revision/Update Information:	This Guide supersedes *Guide to VAX SPM*, Version 3.0
Software Version:	VAX SPM, Version 3.3

Digital Equipment Corporation

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DATATRIEVE
DEC
DEC/CMS
DECgraph
DEC/MMS
DECnet
DECsystem–10
DECsystem–20
DECUS
DECwriter
DIBOL
EduSystem
IAS
MASSBUS
NETPLAN
NETSTART
NETSUPPORT
PDP
PDT
RSTS
RSX
UNIBUS
VAX
VAXcluster
VMS
VT

digital™

Contents

EXAMPLES

FIGURES

TABLES

Preface

This guide explains how to use VAX SPM (Software Performance Monitor) for system tuning and historical reporting. As you become familiar with VAX SPM, refer to the *VAX SPM V3.3 Reference Manual* (AA-R580D-TE) for more detailed descriptions of VAX SPM commands, reports, and display graphics.

Intended Audience

This guide is intended as an introduction to VAX SPM for system managers, programmers, or anyone interested in VAX/VMS system performance.

Version Information

Version 3.3 of VAX SPM runs only on the VAX/VMS Version 5.0 (or later) operating system.

Structure of This Guide

This guide is organized into three areas:

- Getting Started
- Evaluating Performance
- Historical Reporting

Table 1 shows which chapters or appendixes in this guide pertain to the three areas. If a chapter pertains to an area, a diamond appears in the area's column for that chapter. For example, there is a diamond in each area's column for Chapter 1. This means that Chapter 1 pertains to all areas.

Table 1 Overview of Guide's Organization

Chapter/ Appendix	Getting Started	Evaluating Performance	Historical Reporting
Chapter			
1. Introduction	◆	◆	◆
2. Getting Started	◆		
3. Collecting Performance Data	◆	◆	◆
4. Introduction to Reporting	◆	◆	
5. Introduction to System Tuning	◆	◆	
6. Starting System Tuning	◆	◆	
7. Diagnosing a Resource Limitation		◆	
8. Isolating the Resource Limitation		◆	
9. Disk Activity Analysis		◆	
10. Disk Space Analysis		◆	
11. System Program Counter Analysis		◆	
12. Evaluating Performance Using Graphic Displays		◆	
13. Using the Analyzer	◆	◆	
14. Reporting Log File Data	◆	◆	◆
15. Creating and Maintaining History Files		◆	◆
16. Reporting on History File Data		◆	◆
17. Generating Presentation Graphs		◆	◆
18. Reporting Trend Analysis Data			◆
19. The Event Trace Facility		◆	
20. Charging for System Usage		◆	
21. Using VAX SPM on a VAXcluster System		◆	◆
22. Using VAX SPM on Remote Nodes		◆	◆
Appendix			
A. Collector Version of VAX SPM	◆	◆	◆
B. Converting V2.X History Files			◆

Chapter and Appendix Descriptions

- Chapter 1 provides an overview of VAX SPM.
- Chapter 2 guides users through starting and stopping VAX SPM collections, generating a system summary graph and a tabular report, and displaying the system load balance display graphic.
- Chapter 3 describes collecting system performance data using VAX SPM commands and the SPM$MANAGER command procedure.
- Chapter 4 introduces reporting.
- Chapter 5 provides an overview of the tuning process and describes VAX SPM tools for system tuning.
- Chapter 6 guides users through getting started tuning their systems.
- Chapter 7 describes the diagnose step of the "hands-on" approach to system tuning.
- Chapter 8 describes the isolate step of the "hands-on" approach to sytem tuning.
- Chapter 9 describes monitoring disk activity using the file activity display.
- Chapter 10 describes evaluating disk utilization using the Disk Space Analysis facililty.
- Chapter 11 descibes evaluating system CPU usage using the System Program Counter (PC) Analysis facility.
- Chapter 12 describes monitoring system performance interactively using the SPM Display Graphic utilities.
- Chapter 13 describes automatically locating a performance bottleneck using the Performance Analyzer.
- Chapter 14 describes reporting on log file data.
- Chapter 15 describes creating and maintaining a history file.
- Chapter 16 describes reporting on history file data.
- Chapter 17 describes generating presentation graphs.
- Chapter 18 provides various examples of reporting trend analysis data.
- Chapter 19 describes how to monitor system events using the Event Trace facility.
- Chapter 20 describes charging for system usage using the VAX SPM Charge facility.

- Chapter 21 provides additional VAXcluster system collecting and archiving information, and discusses how to balance a workload and isolate disk contention on a VAXcluster system.
- Chapter 22 describes how to invoke VAX SPM on remote nodes.
- Appendix A describes the Collector version of VAX SPM and its use with the SPM$MANAGER.COM command procedure.
- Appendix B describes converting V2.X history files to V3.X format.

Conventions Used in This Document

Convention	Meaning
CTRL/x	CTRL/x indicates that you simultaneously press the CTRL key and another key; for example, CTRL/C, CTRL/Y, CTRL/O.

Associated Documents

Manuals that are useful references are:

- *VAX SPM Version 3.3 Release Notes (AA–X088H–TE)*
- *VAX SPM Reference Manual (AA–R580D–TE)*
- *Guide to VAX/VMS Performance Management (AA–LA43A–TE)*
- *VAX/VMS System Services Reference Manual (AA–LA68A–TE)*
- *VAX/VMS Accounting Utility Manual (AA–LA44A–TE)*
- *VAX/VMS Linker Utility Manual (AA–LA62A–TE)*
- *Introduction to VMS System Management (AA-LA24A-TE)*
- *Guide to Maintaining a VMS System (AA-LA34A-TE)*
- *VAX/VMS Installation and Operations (for your system)*
- *VAX/VMS Install Utility Manual (AA–LA29A–TE)*
- *VAX/VMS System Generation Utility Manual (AA–LA30A–TE)*
- *VMS DCL Concepts Manual (AA–LA10A–TE)*
- *VMS Authorize Utility Manual (AA–LA42A–TE)*

1 Introduction

SPM provides tools for performance evaluation and historical reporting of VMS performance information.

This chapter provides the following information:

- Overview of VAX SPM
- Description of the VAX SPM tools to use for collecting data, evaluating performance, and generating reports from a history data base
- Glossary of terms found in this guide

1.1 Overview of VAX SPM

Any mechanical or electrical device prompts the question, "How well does the device perform its appointed tasks?" The more complex the device, the more need there is for tools to record, analyze, and evaluate the device's behavior. The purpose of evaluating a device's behavior is to measure its performance and reveal ways to improve it.

With the introduction of computer systems where processor, memory, and peripheral devices are capable of simultaneous activity, users are concerned with system performance. The advent of multiprogramming and timesharing systems has further increased the demand for tools to measure and evaluate computer performance. VAX SPM is a comprehensive set of tools designed to measure the performance of VMS systems.

VAX SPM collects, reports, and analyzes performance statistics for VMS systems. General performance statistics can be collected on a VAXcluster system or a stand-alone system with single or multiple CPUs. Detailed statistics can be collected on a per-process basis.

The performance statistics collected by VAX SPM may be analyzed automatically or by the "hands-on" approach. Analysis of the performance reports provides information required for system tuning; avoidance of system bottlenecks; and the design, development, and use of effective applications.

Performance statistics collected daily from a single node or VAXcluster system can be consolidated into a history file for reporting and graphing system performance over time.

VAX SPM is designed for use by system managers, system programmers, and application programmers. Many reports are suitable for presentation to managers, particularly to justify future hardware acquisitions.

With VAX SPM, you can collect data and invoke video displays from any node in a VAXcluster system. In addition, you can invoke collections and the Resource and Investigate video displays on nodes that are not part of a VAXcluster system. Chapter 22 describes invoking VAX SPM on remote nodes.

There are two versions of VAX SPM: the standard version with full functionality and the collector version, which only collects data. Appendix A describes the collector version of VAX SPM.

VAX SPM tools fall into three broad categories:

- Tools for collecting performance data
- Tools for evaluating performance data
- Tools for historical reporting of performance data.

1.2 Collecting VMS Performance Data

VAX SPM provides two ways to collect performance data:

- Conventional collection—using the PERFORMANCE COLLECT commands
- Automatic collection—using the SPM$MANAGER.COM command procedure.

Each method is described in the following sections.

1.2.1 Conventional Data Collections

VAX SPM supplies two collection commands:

```
PERFORMANCE COLLECT=TUNE
PERFORMANCE COLLECT=CAPACITY
```

The data collected by these commands is identical and their existence permits you to run two collection processes concurrently.

VAX SPM collects statistics related to CPU, memory, I/O, and VAXcluster system communication.

For multiprocessor systems such as the VAX 8000 and VAX 6000 series, CPU metrics include data for all processors.

The conventional data collection process is described in more detail in Chapter 3.

1.2.2 Automatic Data Collection

The command procedure SPM$MANAGER.COM is a useful tool that automatically collects data each day and stores it in a log file. It evaluates the data and sends VAXmail messages notifying the system manager if threshold values for performance metrics have been exceeded.

When VAX SPM is installed on your system, SPM$MANAGER.COM is placed in the directory SPM$EXAMPLES:[SYSHLP.EXAMPLES.SPM]. Instructions for setting up SPM$MANAGER.COM and descriptions of the symbols it uses are included in the command procedure. The command procedure's flexible design makes it a useful tool for both performance evaluation and historical reporting. You can customize SPM$MANAGER.COM as your collection and reporting needs change.

VAX SPM enables SPM$MANAGER_CONFIGURE.COM to automate the process of setting up the SPM$MANAGER.COM command procedure. Chapter 3 provides detailed instructions for setting up SPM$MANAGER.COM to begin automatic collecting and reporting on your system.

1.3 Performance Evaluation

Initiate a performance evaluation to improve the throughput or productivity of your computer system.

The VAX SPM tools for performance evaluation consist of one or more utilities that perform the following tasks:

- Reporting tuning information
- Graphing tuning informaton
- Analyzing performance information on-line from a log file
- Displaying tuning information on your terminal
- Collecting and reporting system program counter (PC) data
- Displaying file activity
- Analyzing disk space usage
- Tracing internal VMS events
- Charging for system usage

1.3.1 Reporting Tuning Data

Once you have collected tuning data, the following command is available for reporting tuning data:

```
PERFORMANCE REPORT=LOG_FILE
```

The Reporting facility provides the following options for reporting performance metrics in both tabular and graph form:

- Generating working style graphs of system metrics in ANSI format

- Generating presentation graphs of system metrics in ReGIS or sixel format. Presentation graphs can be pie charts, filled line graphs, histograms, or stacked bar graphs
- Selectively inhibiting the generation of graphs based upon performance metrics you define
- Dumping the data used to generate reports and graphs into a file from which custom reports can be made by using a report generator such as DATATRIEVE
- Generating an interval statistics report for each interval within a reporting period
- Generating a final statistic report that represents the combined statistics from all intervals in a reporting period
- Generating tabular report statistics for prime and nonprime hours

  Chapter 4 introduces VAX SPM reporting. Chapter 5 describes how to use VAX SPM tabular reports to start tuning your system, and Chapter 7 and Chapter 8 describe using reports and graphs to evaluate system performance.

1.3.2 Analyzing Performance from a Log File

The following command invokes the Analyzer:

```
PERFORMANCE ANALYZE=LOG_FILE
```

The VAX SPM Performance Analyzer (or Analyzer) applies a series of tests to VAX SPM log files to locate and describe a performance bottleneck. In addition to identifying problem spots, it makes recommendations for correcting the limitations it discovers.

The Analyzer may be run interactively or from a command file. When invoked interactively, it guides you through a session by displaying each test of the analysis. You can request more details for a test, or change the result of a test to explore other potential bottlenecks. At any time during an interactive session, you can choose to allow the analysis to proceed directly to its conclusion.

An important feature of the Analyzer is the access it provides to the threshold values used in its tests. Threshold values can be displayed at any time during a session. They can also be changed and saved for subsequent analysis sessions.

A report can be produced for each analysis session. The report describes the conclusion of the analysis, lists the tests that support the conclusion, and provides recommendations to correct limitations discovered during the analysis. You can run the Analyzer from a command file and receive the session report, conclusion, and recommendations without having to direct the Analyzer.

The Analyzer is valuable to both novice and expert system tuners. The novice can use the Analyzer to diagnose a problem, correct the problem, and verify the improvements in a relatively short time. In addition, a novice can learn about system tuning and using VAX SPM reports while actually tuning the system. Experienced performance analysts can use the Analyzer to confirm their diagnosis, or to change the result of a test during an interactive session and explore other potential system bottlenecks.

Chapter 13 describes the Analyzer and how to use it.

1.3.3 Displaying Performance Information on Your Terminal

The Video Graphics utilities display tuning information on a video screen and allow you to print the displays on a graphics printer. Some displays require a VT240 or other ReGIS-compatible terminal. If the terminal supports color, a multicolored display is generated.

The commands used to invoke the video displays are:

```
PERFORMANCE DISPLAY=INVESTIGATE
PERFORMANCE DISPLAY=RESOURCE
```

There are two types of displays: Investigate, which requires a ReGIS-compatible terminal for most displays; and Resource, which displays on any ANSI-compatible terminal.

The Investigate displays show system metrics including memory, I/O, and CPU data. They are used for interactive performance evaluation. The Resource displays are used for VAXcluster system analysis and show resource usage for a maximum of eight nodes or disks in a cluster per display.

Two modes of display are possible: real-time, which shows the performance of the running system and playback (for Investigate only), which reads and displays the contents of a log or history file.

Chapter 12 describes the video displays.

1.3.4 Collecting and Reporting System Program Counter Data

The following commands collect and report on system program counter (PC) data:

```
PERFORMANCE COLLECT=SYSTEM_PC
PERFORMANCE REPORT=SYSTEM_PC
```

Use the VAX SPM System PC facility when performance reports indicate that system activity is high to determine where system time is being spent.

The VAX SPM System PC facility samples program counter values as the system runs. It uses these samples to report on processor use by executive image, module, process (by processor mode), and by interrupt priority level (IPL). Reports can be generated by using a filter to examine the activity of a specific process or the interrupt stack.

Chapter 11 describes collecting and reporting system PC data.

1.3.5 Analyzing File Activity

The command used to invoke the file activity display is:

```
PERFORMANCE DISPLAY=FILES
```

The File Activity utility displays file activity on your terminal. It helps to identify the files with the highest read/write rate for the disks you specify. Use the file activity displays when a performance investigation indicates that excessive disk activity may be the cause of poor system performance.

1.3.6 Analyzing Disk Space

The command used to report on ODS-2 (On-Disk Structure Level 2) disk volumes is:

```
PERFORMANCE REPORT=DISK_SPACE
```

VAX SPM provides detailed analysis of ODS-2 disk volumes.

The VAX SPM Disk Space facility collects volume attribute data, free space utilization, and allocated space utilization, and produces a disk space report in an ASCII file. Analysis of this report can detect the need for data compression using the Backup facility as well as inconsistencies between volume use and volume initialization values.

Chapter 10 describes the Disk Space report.

1.3.7 Tracing Internal VMS Events

The Event Trace facility (ETF) provides the knowledgeable VMS system programmer with the ability to access VMS Executive data that VMS and VAX SPM do not routinely provide. For example, ETF could be used to determine the number of times a critical routine is called in a user-written device driver.

Use ETF to write a trace monitor that calls event trace service routines. These routines define locations in system virtual address space as trace points to be monitored. When one of these trace points is reached, a user-written data collection routine is invoked to collect data.

Chapter 19 describes the Event Trace facility.

1.3.8 Charging for System Usage

The command used to generate a Charge report is:

```
PERFORMANCE REPORT=CHARGE
```

The VAX SPM Charge facility reads VMS accounting data and applies a set of user-defined monetary values for various system resources. The result is a report showing a detailed breakdown of charges by user name, account name, UIC, job type, and individual job or process. This information can be used for detecting resource bottlenecks or as the basis for billing.

Chapter 12 describes charging for system usage.

1.4 Historical Reporting

Historical reporting can be initiated as part of performance evaluation, and continued once a system has been tuned. As part of performance evaluation, historical reports and graphs can be used to establish a performance baseline or to verify the results of performance adjustments. Once a system has been tuned, historical graphs and reports can be used to justify the purchase of additional equipment, pinpoint underutilized resources, or define trends in system performance or use of resources.

Data collected each day for a single node or VAXcluster system can be consolidated into a historical data base called a history file. Data in this history file can be accessed to produce reports and graphs summarizing system performance and resource utilization over time.

Data in the history file cannot be analyzed by the VAX SPM Analyzer.

The tools for historical reporting consist of utilities that perform the following tasks:

- Archiving daily log files into a history data base composed of:
    - Consolidated daily log file data
    - Consolidated VAXcluster system data
- Generating graphs and reports of historical information
- Displaying historical information on your terminal.

1.4.1 Archiving Daily Log Files

The following DCL command archives data from a log file to create a historical data base called a history file:

```
PERFORMANCE ARCHIVE=HISTORY
```

Use the VAX SPM Archive facility to create and maintain a history file from daily log files. Consolidating daily log files into a history file conserves disk space while accumulating performance data for historical reporting.

If you are running VAX SPM on a VAXcluster system, archiving daily log files from all nodes conserves disk space and facilitates evaluation and reporting of cluster performance. The VAX SPM Archive facility enables you to add, replace, and delete data in the history file. It also provides a way to review the contents by listing information in the history file.

Chapter 15 describes the Archive facility.

1.4.2 Generating Graphs and Reports

The following command generates tabular reports and graphs from a history file:

```
PERFORMANCE REPORT=HISTORY history-file-spec
```

The VAX SPM Historical Reporting facility provides the following options for generating reports and graphs:

- Generate working style graphs of resource utilization in ANSI format
- Generate presentation graphs of resource utilization in ReGIS or sixel format. Presentation graphs may be pie charts, filled line graphs, histograms or stacked bar graphs
- Generate graphs or reports for a single node or for a VAXcluster system
- Limit graphs and reports to relevant hours and days by defining holidays, and prime and nonprime hours and days
- Select a system resource such as memory, I/O, and CPU for reporting or graphing utilization over time
- Select a performance metric such as balance set, file opens, or memory utilization for graphing or reporting over time
- Average data over a period of time to graph the typical day, week, or month
- Graph data from selected devices or disks
- Generate graphs and reports for reporting units of a day, week, or month, or a user-defined unit such as a fiscal month or quarter
- Produce dump files in ASCII or binary format from which custom reports or graphs can be generated by using VAX DATATRIEVE or VAX-11 DECgraph

Chapter 4 introduces VAX SPM reporting. Chapter 16 describes the HISTORY file reporting command. Chapter 18 provides examples applying the commands described in Chapter 16 to produce a number of trend analysis reports and graphs.

1.4.3 Displaying Historical Information

The following command displays historical information on your terminal:

```
PERFORMANCE DISPLAY=INVESTIGATE/INPUT=history-file.dat
```

The Investigate display requires a ReGIS-compatible terminal. It shows system metrics including memory, I/O, and CPU information. The Investigate display reads the contents of a history file and presents statistics in visual, dynamic form.

Chapter 12 describes the Investigate video displays.

1.5 Term Definitions

Throughout this guide a variety of terms are used in relation to time. This section defines time-related terms and the PERFORMANCE commands, and the qualifiers and keywords used to specify them.

Analysis Period

A measure of time within the collection period selected for analysis. It can be the entire collection period but usually is a portion when system versus task CPU activity is high. May also refer to the records of data collected within this time.

Specified using the /BEFORE and /SINCE qualifiers or HISTOGRAM interactive command of PERFORMANCE ANALYZE=LOG_FILE command.

Archive Interval

A measure of time indicating the collection frequency represented by records in a history file. Log file records are usually archived at a longer interval than they are collected. When the archive interval of the history file is longer than the collection period of the log file, data is averaged over the longer archive interval.

Specified using the /INTERVAL qualifier of PERFORMANCE ARCHIVE=HISTORY command.

Archive Period

A measure of time within the log file collection period selected for archiving into the history file. It can be the entire collection period of the log file. Also refers to the records of data collected within this time.

Specified using the /BEFORE and /SINCE qualifiers with the PERFORMANCE ARCHIVE=HISTORY command.

Bucket or Time Bucket

A time unit defined by the value of the ARCHIVE=HISTORY qualifier /INTERVAL, which establishes the structure for storing records in a history file. The creation and addition of records to buckets may be observed by specifying /LOG for history file transactions.

Specified using the /INTERVAL qualifier of the PERFORMANCE ARCHIVE=HISTORY command.

Collection Interval or Sample Interval

A measure of time indicating the frequency with which records are collected and written to the log file.

Specified using the /INTERVAL qualifier of the PERFORMANCE COLLECT=TUNE and PERFORMANCE COLLECT=CAPACITY commands.

Collection Period

A measure of time starting when the collection process begins and ending when it stops collecting data. Also refers to all records of data collected within this time.

Specified using the /BEGIN and /END qualifiers with PERFORMANCE COLLECT=TUNE or PERFORMANCE COLLECT=CAPACITY commands.

Reporting Interval

A measure of time indicating the frequency with which records are reported from a history file or a log file. If the reporting interval is longer than the collection or archive interval, data is prorated over the longer reporting interval. If the reporting interval is shorter than the collection or archive interval, meaningless data may result. For final tabular reports, reporting intervals are computed together; for interval tabular reports, they are reported individually. Each reporting interval in the reporting period is represented by a column on a graph.

Specified using the /INTERVAL qualifier for the PERFORMANCE REPORT=LOG_FILE command. For the PERFORMANCE REPORT=HISTORY command, specified directly using the INTERVAL or REPORT_INTERVAL keyword of the /GENERAL qualifier and indirectly according to the reporting unit.

Reporting Period

A measure of time indicating the portion of the log or history file that is reported. If no value is specified, the entire log or history file is reported.

Specified using the /BEFORE and /SINCE qualifiers for the REPORT=LOG_FILE and REPORT=HISTORY commands.

Reporting Unit

A measure of time for which history file statistics are reported or graphed. A reporting unit can be a day, week, or month, or a user-defined value. For tabular reports, a final report is generated for each reporting unit within the reporting period and an interval report is generated for each interval within the reporting period. For graphs, a graph is generated for each reporting unit within the reporting period. Each column in the graph represents one interval within the reporting unit. The interval size is determined by the reporting unit specified.

Specified using the /DAILY, /WEEKLY, and /MONTHLY or the /GENERAL qualifiers for the PERFORMANCE REPORT=HISTORY command.

2 Getting Started

The purpose of this chapter is to familiarize users with a few basic VAX SPM activities. After performing these activities, proceed to the relevant chapters in this guide and in the *VAX SPM Reference Manual* for more detail.

This chapter describes how to perform the following SPM activities:

- Set privileges for running VAX SPM.
- Determine if VAX SPM collections are running on your system.
- Collect performance data.
- Generate a system summary graph.
- Generate a tabular report.
- Show the Load Balance video display on your terminal.

2.1 Prerequisite Privileges

To begin collecting performance data, your account requires either the SETPRV privilege or all the following privileges:

```
ALTPRI    CMKRNL    DETACH    PRMMBX    PSWAPM    NETMBX
SYSLCK    SYSNAM    SYSPRV    TMPMBX    WORLD
```

2.2 Determining Whether Collections are Running

VAX SPM collection commands create detached collection processes named SPM_TUNE and SPM_CAPACITY. Only one SPM_TUNE and one SPM_CAPACITY process can run simultaneously on a given node.

To determine if a tuning collection process is running on your current node, type the following command:

```
$ PERFORMANCE COLLECT=TUNE/INQUIRE
```

If a tuning collection process is not already running, VAX SPM displays the following message:

```
%SPM-E-CONORU, collections not running
```

If this message displays, proceed to Section 2.3 to start collections.

If a tuning process is already running on your current node, VAX SPM displays a status report for the collection process similar to the Status report displayed in Figure 2–1.

Figure 2–1 VAX SPM Collection Process Status Report

```
$ PERFORMANCE COLLECT=TUNE/INQUIRE
       Status of SPM_TUNE
❶    Logfile                : USR$:[USER_A]SPM$COLLECT_TUNE.DAT;1
❷    Sample Interval        : 300 seconds
❸    Current Sample         : 5
❹    Collections Beginning :  1-FEB-1988 11:40:42.87
❺    Collections Ending     : not specified
❻    Collecting CPU Statistics
           Collecting Disk Statistics
    Collecting I/O Statistics
    Collecting Memory Statistics
    Collecting Page Fault Statistics
    Collecting Server Statistics
    Collecting XQP Statistics
```

The following describes numbered items in the Status report:

❶ Name of the log file created by the collection process

❷ Length of the sample interval

❸ Number of the current sample being collected

❹ Begin time for the collection process

❺ End time for the collection process (none was specified for this collection)

❻ List of the classes of data being collected

To determine if a capacity collection process is running on your current node, type the following command:

```
$ PERFORMANCE COLLECT=CAPACITY/INQUIRE
```

If a capacity collection process is not already running, VAX SPM displays the following message:

```
%SPM-E-CONORU, collections not running
```

If this message displays, proceed to Section 2.3 and start collections.

If a capacity collection process is running on your system, VAX SPM displays a Status report similar to the one shown in Figure 2-1.

If both tune and capacity collections are running on your system, you must either stop one or wait until one is finished before you can begin a collection process. Ask your system manager when one of the collection processes is due to stop, or ask for the name of a log file you can use for reporting.

2.3 Collecting Performance Data

VAX SPM provides two commands to begin the collection process: PERFORMANCE COLLECT=TUNE and PERFORMANCE COLLECT=CAPACITY (Figure 2–2). The PERFORMANCE COLLECT=TUNE command creates the collection process SPM_TUNE, and the PERFORMANCE COLLECT=CAPACITY command creates the collection process SPM_CAPACITY. These commands may be used interchangeably because they both collect the same data. The purpose of having two commands is to permit the simultaneous collection of data at different collection intervals. For example, tuning data would be at a short interval and historical reporting data would be at a longer interval. Chapter 3 provides more information about these commands.

Because VAX SPM permits only one tune and one capacity collection process to run at a time, first determine whether collections are running as shown in the previous section.

If capacity collections are already running on your system, type the following command to begin collecting performance data:

```
$ PERFORMANCE COLLECT=TUNE/CLASS=(ANALYZE)/OUTPUT=SPM$COLLECT_TUNE.DAT
```

If tuning collections are already running on your system, type the following command to begin collecting performance data:

```
$ PERFORMANCE COLLECT=CAPACITY/CLASS=(ANALYZE)/OUTPUT=SPM$COLLECT_CAPACITY.D
```

You may verify that the collection process has begun by typing the following command:

```
$ PERFORMANCE COLLECT=[ TUNE     ] /INQUIRE
                      [ CAPACITY ]
```

Allow the process to collect performance data for at least 30 minutes.

Type the following command to stop the collection process and close the log file SPM$COLLECT_TUNE.DAT or SPM$COLLECT_CAPACITY.DAT:

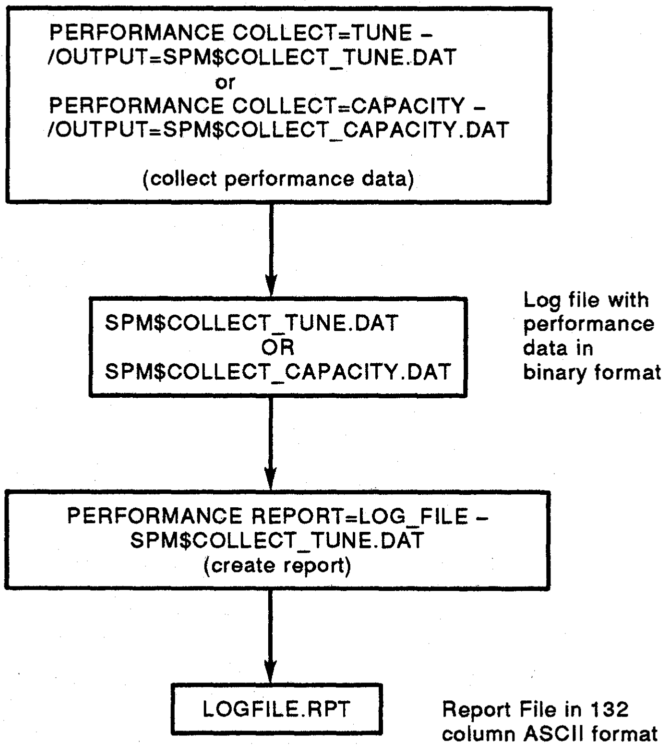
```
$ PERFORMANCE COLLECT=[ TUNE     ] /STOP
                      [ CAPACITY ]
```

A variety of reports may be generated from this log file. Section 2.4 describes how to generate the system summary graph and Section 2.5 describes how to generate the tabular report.

2.4 Generating a System Summary Graph

Figure 2–2 shows that both VAX SPM collection processes (tune and capacity) create a log file of VAX SPM data. Note that the log file is in a format that can be read only by VAX SPM. You may convert a log file into a file that can be edited and read by using VAX DATATRIEVE and VAX-11 DECgraph. Chapter 16 describes this procedure.

Figure 2–2 Collecting and Reporting Performance Data



```
PERFORMANCE COLLECT=TUNE -
/OUTPUT=SPM$COLLECT_TUNE.DAT
or
PERFORMANCE COLLECT=CAPACITY -
/OUTPUT=SPM$COLLECT_CAPACITY.DAT
```

(collect performance data)
SPM$COLLECT_TUNE.DAT
OR
SPM$COLLECT_CAPACITY.DAT
Log file with performance data in binary format

```
PERFORMANCE REPORT=LOG_FILE -
SPM$COLLECT_TUNE.DAT
```

(create report)
LOGFILE.RPT
Report File in 132 column ASCII format

The VAX SPM report command reads the log file and creates report files. These report files are in 132-column ASCII format and may be edited, printed, or typed on the screen.

The system summary graph represents system activity and provides a overview of system performance. It is a good place to start when evaluating performance on your system. You can use the system summary graph to identify times of high system activity for further investigation. Type the following command to generate a system summary graph:

```
$ PERFORMANCE REPORT=LOG_FILE/NOTABULAR/GRAPH=SUMMARY -
  /OUTPUT=SUMMARY.RPT SPM$COLLECT.DAT
```

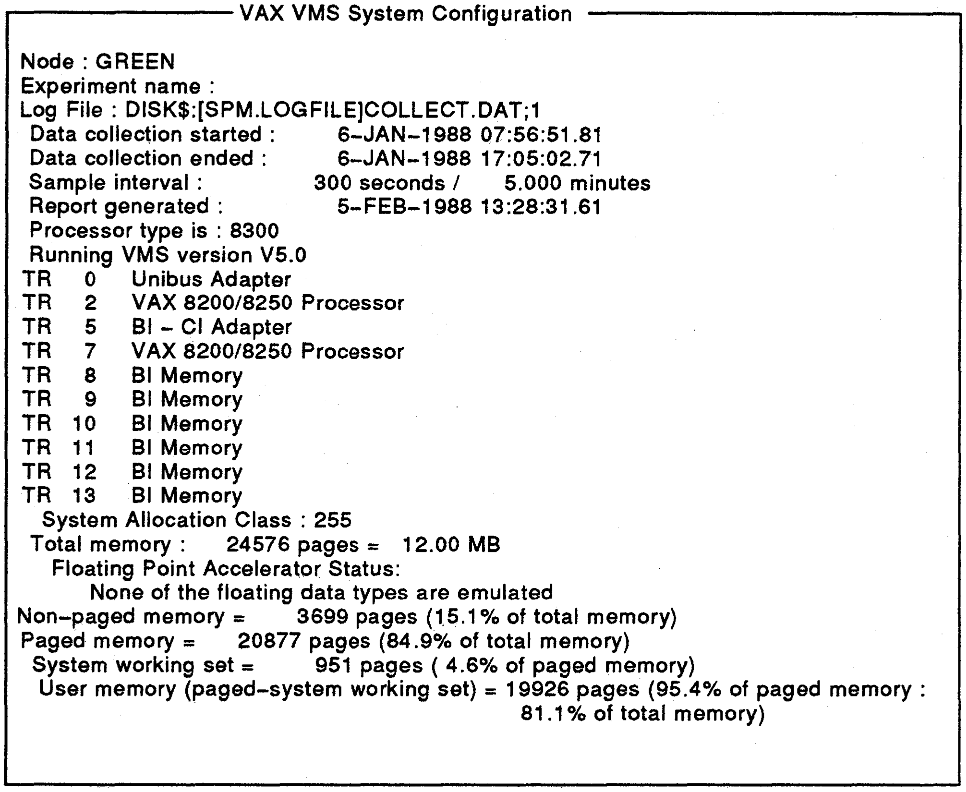
For a hard copy of the system summary graph, print the file SUMMARY.RPT. The report is in 132-column format and must be printed on an appropriate printer.

The System Summary report contains three parts: the system configuration, the system summary graph, and the run statistics.

System Configuration: The first part of the Summary report is the VMS system configuration. Figure 2–3 is an example of the System Configuration report.

The System Configuration report provides information about the collection process, and the VAX system hardware and operating system.

Figure 2–3 VAX SPM System Configuration Report



```
-------------- VAX VMS System Configuration --------------

Node : GREEN
Experiment name :
Log File : DISK$:[SPM.LOGFILE]COLLECT.DAT;1
 Data collection started :     6-JAN-1988 07:56:51.81
 Data collection ended :       6-JAN-1988 17:05:02.71
 Sample interval :           300 seconds /    5.000 minutes
 Report generated :            5-FEB-1988 13:28:31.61
 Processor type is : 8300
 Running VMS version V5.0
TR   0   Unibus Adapter
TR   2   VAX 8200/8250 Processor
TR   5   BI - CI Adapter
TR   7   VAX 8200/8250 Processor
TR   8   BI Memory
TR   9   BI Memory
TR  10   BI Memory
TR  11   BI Memory
TR  12   BI Memory
TR  13   BI Memory
  System Allocation Class : 255
 Total memory :    24576 pages =   12.00 MB
   Floating Point Accelerator Status:
      None of the floating data types are emulated
Non-paged memory =      3699 pages (15.1% of total memory)
Paged memory =     20877 pages (84.9% of total memory)
 System working set =       951 pages ( 4.6% of paged memory)
  User memory (paged-system working set) = 19926 pages (95.4% of paged memory :
                                                81.1% of total memory)
```

System Summary Graph: The second part of the report is the system summary graph. The system summary graph displays the type and percentage of system activity over a collection period. The graph is made up of columns that contain the characters *, C, and I. Each column represents a collection interval. The characters in each column represent the type of system activity occurring during the interval. A legend describing the type of system activity represented by each character appears at the bottom of the graph.

Figure 2–4 shows a sample system summary graph for a 30-minute collection period. Note that this graph has fewer columns than a report for an 8-hour collection period.

Figure 2–4 System Summary Graph for a 30-Minute Collection Period

```
Report Date: 14-MAR-1989 18:21                                  GREEN                              VAX SPM V3.3         Page     2
                                                System Summary (Percent) vs. Time of Day
      From:   8-NOV-1988 15:52:20.32          Each column = 300 seconds /    5.00 minutes         To   :  8-NOV-1988 16:22:29.96
      -+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-----
 100% + I   I                                                                                                                  +
      ! II *I                                                                                                                  !
      ! ** *I                                                                                                                  !
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
  90% + *****                                                                                                                  +
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
  80% + *****                                                                                                                  +
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
  70% + *****                                                                                                                  +
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
      ! *****                                                                                                                  !
  60% + ****C                                                                                                                  +
      !I****C                                                                                                                  !
      !I*C**C                                                                                                                  !
      !ICC**C                                                                                                                  !
      !ICC**C                                                                                                                  !
  50% +ICC**C                                                                                                                  +
      !ICCCCC                                                                                                                  !
      !*CCCCC                                                                                                                  !
      !*CCCCC                                                                                                                  !
      !*CCCCC                                                                                                                  !
  40% +*CCCCC                                                                                                                  +
      !*CCCCC                                                                                                                  !
      !*CCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
  30% +CCCCCC                                                                                                                  +
      !CCCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
  20% +CCCCCC                                                                                                                  +
      !CCCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
  10% +CCCCCC                                                                                                                  +
      !CCCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
      !CCCCCC                                                                                                                  !
      !CCCCCC..................................................................................................................!
      -+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-----
     15:52
          16:17
                   "C" = CPU Only          "*" = CPU & I/O          "I" = I/O Only           "." = Data Unavailable
```

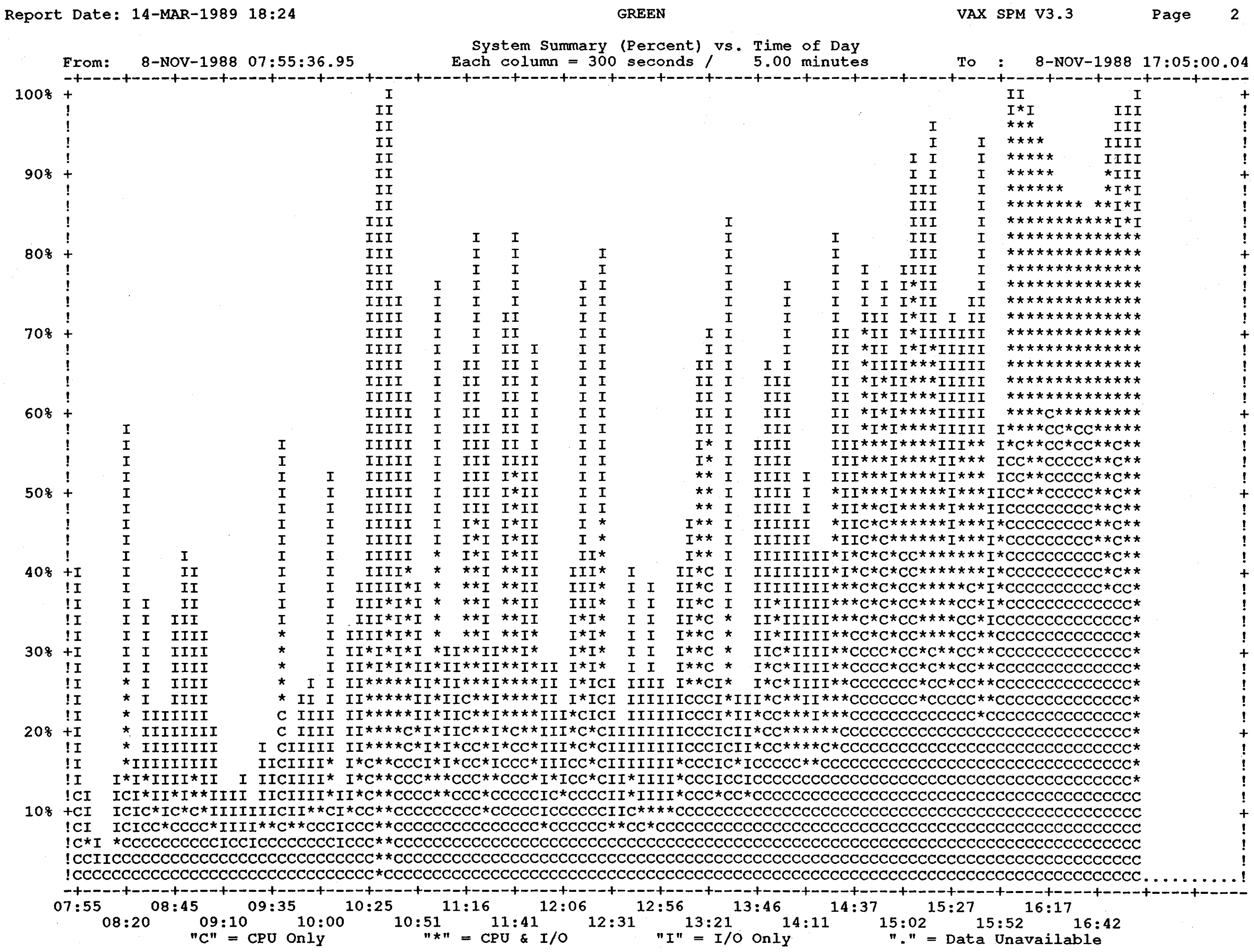
Run Statistics: The third page of the report contains general information about the reporting process. The run statistics provide information about the collection and reporting parameters used to generate the report. The command line repeats the DCL command that generated the report. The last line of the report labeled ELAPSED shows the amount of system resources used to generate the report. Figure 2–5 is an example of the run statistics page.

Figure 2–5 VAX SPM Run Statistics Report

```
Run Statistics
--------------

Total Intervals Found For Analysis: 110
Prime Hours: 08, 09, 10, 11, 12, 13, 14, 15, 16
Input Source: Log File V3.3

Command Line:

REPORT=LOG_FILE/NOTABULAR/GRAPH=(SUMMARY)/OUTPUT=SUMMARY.RPT
DISK$:[SPM.LOGFILE]COLLECT.DAT

ELAPSED: 0 00:00:09.17  CPU: 0:00:03.22  BUFIO: 7  DIRIO: 184  FAULTS: 836
```

The system summary graph shows the times of high system activity, and whether it was CPU, I/O, or both CPU and I/O activity. Figure 2–6 is a system summary graph for an 8-hour collection period. The time segments whose columns approach the top of the chart represent a time of peak usage on this graph.

Figure 2–6 System Summary Graph for an 8-Hour Collection Period



```
Report Date: 14-MAR-1989 18:24                                GREEN                              VAX SPM V3.3        Page    2

                                               System Summary (Percent) vs. Time of Day
     From:   8-NOV-1988 07:55:36.95          Each column = 300 seconds /    5.00 minutes         To  :   8-NOV-1988 17:05:00.04
     -+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-----
100% +                                I                                                               II           I          +
     !                               II                                                               I*I        III          !
     !                               II                                                       I       ***        III          !
     !                               II                                                       I    I  ****      IIII          !
     !                               II                                                     I I    I  *****     IIII          !
 90% +                               II                                                     I I    I  *****     *III          +
     !                               II                                                     III    I  ******    *I*I          !
     !                               II                                                     III    I  ******** **I*I          !
     !                              III                                  I                  III    I  ***********I*I          !
     !                              III        I   I                     I          I       III    I  **************          !
 80% +                              III        I   I        I            I          I       III    I  **************          +
     !                              III        I   I        I            I          I  I   IIII    I  **************          !
     !                              III    I   I   I      I I            I     I    I  I I I*II    I  **************          !
     !                              IIII   I   I   I      I I            I     I    I  I I I*II   II  **************          !
     !                              IIII   I   I  II      I I            I     I    I  III I*II I II  **************          !
 70% +                              IIII   I   I  II      I I          I I     I    II *II I*IIIIIII  **************          +
     !                              IIII   I   I  II I    I I          I I     I    II *II I*I*IIIII  **************          !
     !                              IIII   I  II  II I    I I         II I   I I    II *IIII***IIIII  **************          !
     !                              IIII   I  II  II I    I I         II I   III    II *I*II***IIIII  **************          !
     !                              IIIII  I  II  II I    I I         II I   III    II *I*II***IIIII  **************          !
 60% +                              IIIII  I  II  II I    I I         II I   III    II *I*I****IIIII  ****C*********          +
     !     I                        IIIII  I  III II I    I I         II I   III    II *I*I****IIIII I****CC*CC*****          !
     !     I               I        IIIII  I  III II I    I I         I* I  IIII    III***I****III** I*C**CC*CC**C**          !
     !     I               I        IIIII  I  III IIII    I I         I* I  IIII    III***I****II*** ICC**CCCCC**C**          !
     !     I               I    I   IIIII  I  III I*II    I I         ** I  IIII I  III***I****II*** ICC**CCCCC**C**          !
 50% +     I               I    I   IIIII  I  III I*II    I I         ** I  IIII I  *II***I*****I***IICC**CCCCC**C**          +
     !     I               I    I   IIIII  I  III I*II    I I         ** I  IIII I  *II**CI*****I***IICCCCCCCCC**C**          !
     !     I               I    I   IIIII  I  I*I I*II    I *        I** I  IIIIII  *IIC*C******I***I*CCCCCCCCC**C**          !
     !     I               I    I   IIIII  I  I*I I*II    I *        I** I  IIIIII  *IIC*C******I***I*CCCCCCCCC**C**          !
     !     I     I         I    I   IIIII  *  I*I I*II    II*        I** I  IIIIIIII*I*C*C*CC*******I*CCCCCCCCCC*C**          !
 40% +I    I     II        I    I   IIII*  *  **I **II   III*  I    II*C I  IIIIIIII*I*C*C*CC*******I*CCCCCCCCCC*C**          +
     !I    I     II        I    I  IIIII*I *  **I **II   III*  I I  II*C I  IIIIIIII***C*C*CC*****C*I*CCCCCCCCCC*CC*          !
     !I    I I   II        I    I  III*I*I *  **I **II   III*  I I  II*C I  II*IIIII***C*C*CC****CC*I*CCCCCCCCCCCCC*          !
     !I    I I  III        I    I  III*I*I *  **I **II   I*I*  I I  II*C *  II*IIIII***C*C*CC****CC*ICCCCCCCCCCCCCC*          !
     !I    I I  IIII       *    I IIII*I*I *  **I **I*   I*I*  I I  I**C *  II*IIIII**CC*C*CC****CC**CCCCCCCCCCCCCC*          !
 30% +I    I I  IIII       *    I II*I*I*I *II**II**I*   I*I*  I I  I**C *  IIC*IIII**CCCC*CC*C**CC**CCCCCCCCCCCCCC*          +
     !I    I I  IIII       *    I II*I*I*II*II**II**I*II I*I*  I I  I**C *  I*C*IIII**CCCC*CC*C**CC**CCCCCCCCCCCCCC*          !
     !I    * I  IIII       *  I I II*****II*II***I****II I*ICI IIII I**CI*  I*C*IIII**CCCCCCC*CC*CC**CCCCCCCCCCCCCC*          !
     !I    * I  IIII       * II I II*****II*IIC**I****II I*ICI IIIIIICCCI*III*C**II***CCCCCCC*CCCCC**CCCCCCCCCCCCCC*          !
     !I    * IIIIIII       C IIII II*****II*IIC**I****III*CICI IIIIIICCCI*II*CC***I***CCCCCCCCCCCCC*CCCCCCCCCCCCCCC*          !
 20% +I    * IIIIIIII      C IIII II****C*I*IIC**I*C**III*C*CIIIIIIIICCCICII*CC******CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC*          +
     !I    * IIIIIIII    I CIIIII II****C*I*I*CC*I*CC*III*C*CIIIIIIIICCCICII*CC****C*CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC*          !
     !I    *IIIIIIIII    IICIIII* I*C**CCCI*I*CC*ICCC*IIICC*CIIIIIII*CCCIC*ICCCCC**CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC*          !
     !I   I*I*IIII*II  I IICIIII* I*C**CCC***CCC**CCC*I*ICC*CII*IIII*CCCICCICCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC*          !
     !CI  ICI*II*I**IIII IICIIII*II*C**CCCC**CCC*CCCCCIC*CCCCII*IIII*CCC*CC*CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC          !
 10% +CI  ICIC*IC*C*IIIIIIICII**CI*CC**CCCCCCCCC*CCCCCICCCCCCIIC****CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC          +
     !CI  ICICC*CCCC*IIII**C**CCCICCC**CCCCCCCCCCCCCCC*CCCCCC**CC*CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC          !
     !C*I *CCCCCCCCCICCICCCCCCCCCICCC**CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC          !
     !CCIICCCCCCCCCCCCCCCCCCCCCCCCCCC**CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC          !
     !CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC*CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC..........!
     -+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-----
    07:55     08:45     09:35     10:25     11:16     12:06     12:56     13:46     14:37     15:27     16:17
         08:20     09:10     10:00     10:51     11:41     12:31     13:21     14:11     15:02     15:52     16:42
                  "C" = CPU Only          "*" = CPU & I/O         "I" = I/O Only          "." = Data Unavailable
```

2.5 Generating a Tabular Report

The tabular report provides detailed information about a variety of system resources. Most of the information needed to diagnose and isolate resource limitations is in the tabular report.

A tabular report may be generated for an entire collection period covered by the log file or for periods of interest to further investigate the causes of high resource use. The /BEFORE and /SINCE qualifiers are used to specify a period of interest.

Type the following command to generate a tabular report for an entire collection period:

```
$ PERFORMANCE REPORT=LOG_FILE/TABULAR/NOGRAPH/OUTPUT=TABULAR.RPT-
  SPM$COLLECT.DAT
```

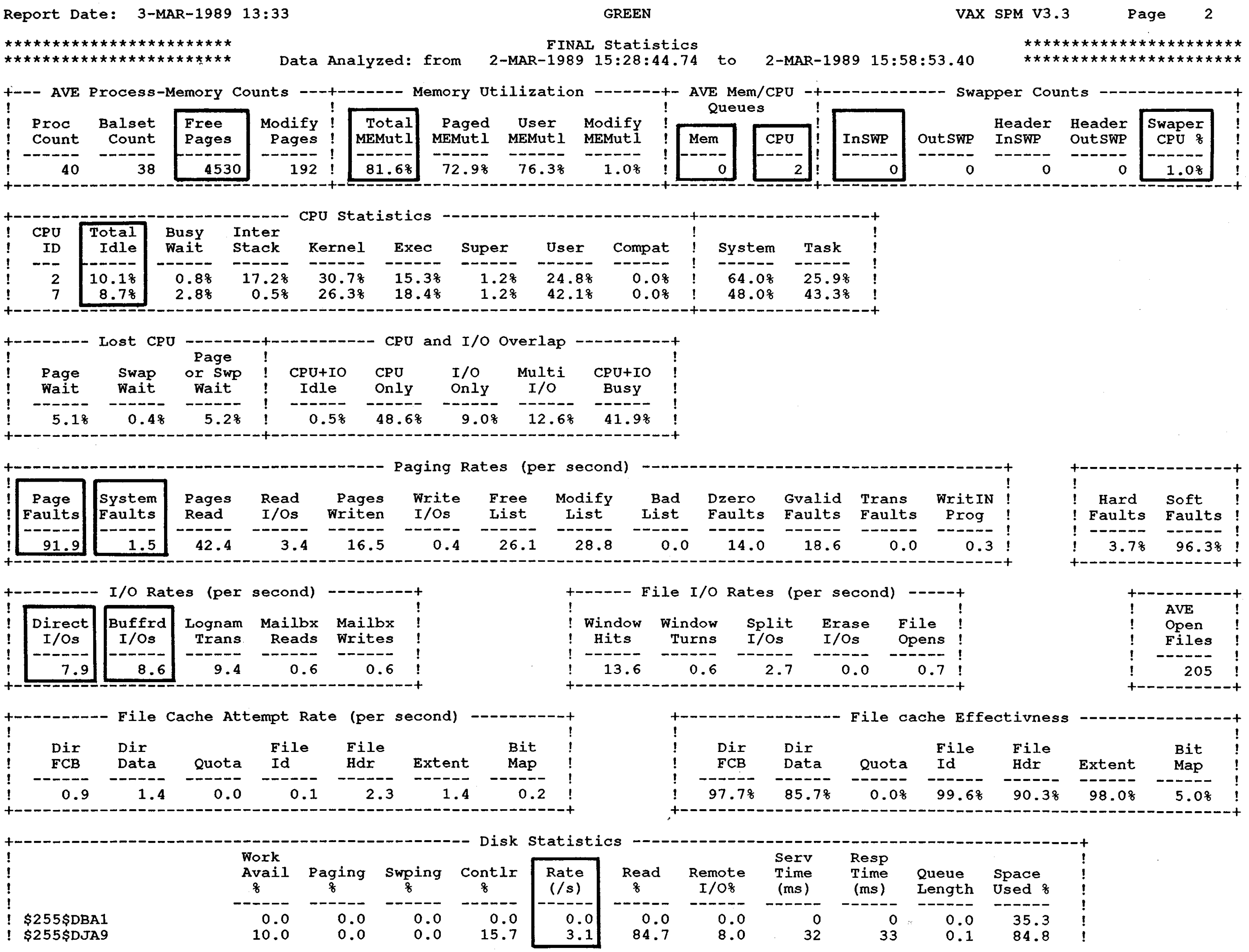
This command gives final statistics in which values for the reported metrics are given over the collection period covered by the log file.

For hard copy of the tabular report, print the file TABULAR.RPT. The report is in 132-column format and must be printed on an appropriate printer.

The tabular report is preceded and followed by VMS system configuration and run statistics similar to those in the System Summary report.

Figure 2–7 is an example of a tabular report. Values of reported metrics are given over a 30-minute collection period.

Table 2–1 lists the fields of the tabular report used to diagnose memory, I/O, and CPU limitations.



```
Report Date:  3-MAR-1989 13:33                         GREEN                            VAX SPM V3.3      Page    2

************************                            FINAL Statistics                                 ***********************
************************     Data Analyzed: from    2-MAR-1989 15:28:44.74  to    2-MAR-1989 15:58:53.40     ***********************

+--- AVE Process-Memory Counts ---+------- Memory Utilization -------+- AVE Mem/CPU -+------------- Swapper Counts --------------+
!                                 !                                   !    Queues     !                                            !
!  Proc   Balset   Free   Modify  !  Total   Paged   User    Modify   !               !                  Header  Header   Swaper    !
!  Count   Count   Pages   Pages  ! MEMutl  MEMutl  MEMutl  MEMutl    !  Mem     CPU  !  InSWP  OutSWP   InSWP   OutSWP   CPU %     !
!  ------  ------  ------  ------ ! ------  ------  ------  ------    !  -----  ----- !  ------ ------   ------  ------   ------    !
!    40      38    4530      192  !  81.6%   72.9%   76.3%    1.0%    !     0      2  !      0      0       0       0     1.0%     !
+---------------------------------+-----------------------------------+---------------+--------------------------------------------+

+-------------------------- CPU Statistics --------------------------+-----------------+
!  CPU   Total   Busy   Inter                                        !                 !
!  ID    Idle    Wait   Stack   Kernel   Exec   Super   User  Compat !  System   Task  !
!  ---   ------  ------ ------  ------  ------  ------  ------ ------ !  ------  ------ !
!   2    10.1%   0.8%   17.2%   30.7%   15.3%    1.2%   24.8%   0.0% !   64.0%   25.9% !
!   7     8.7%   2.8%    0.5%   26.3%   18.4%    1.2%   42.1%   0.0% !   48.0%   43.3% !
+--------------------------------------------------------------------+-----------------+

+-------- Lost CPU --------+----------- CPU and I/O Overlap ----------+
!                   Page   !                                          !
!   Page    Swap   or Swp  !  CPU+IO   CPU     I/O    Multi   CPU+IO  !
!   Wait    Wait    Wait   !   Idle    Only    Only    I/O     Busy   !
!  ------  ------  ------  !  ------  ------  ------  ------  ------  !
!    5.1%    0.4%    5.2%  !    0.5%   48.6%    9.0%   12.6%   41.9%  !
+--------------------------+------------------------------------------+

+------------------------------------ Paging Rates (per second) ------------------------------------+   +----------------+
!                                                                                                    !   !                !
!  Page   System  Pages   Read   Pages   Write   Free   Modify   Bad   Dzero  Gvalid  Trans  WritIN  !   !  Hard    Soft  !
! Faults  Faults  Read    I/Os   Writen  I/Os    List    List    List  Faults Faults  Faults  Prog   !   ! Faults  Faults !
! ------  ------  ------  ------ ------  ------  ------  ------  ------ ------ ------  ------ ------  !   ! ------  ------ !
!  91.9     1.5    42.4    3.4    16.5    0.4    26.1    28.8    0.0   14.0    18.6    0.0    0.3   !   !   3.7%   96.3% !
+----------------------------------------------------------------------------------------------------+   +----------------+

+--------- I/O Rates (per second) ---------+           +------ File I/O Rates (per second) -----+            +----------+
!                                          !           !                                        !            !   AVE    !
! Direct  Buffrd  Lognam  Mailbx  Mailbx   !           ! Window  Window  Split   Erase   File   !            !   Open   !
!  I/Os    I/Os   Trans   Reads   Writes   !           !  Hits   Turns   I/Os    I/Os    Opens  !            !   Files  !
! ------  ------  ------  ------  ------   !           ! ------  ------  ------  ------  ------ !            !  ------  !
!   7.9     8.6     9.4     0.6     0.6    !           !  13.6    0.6     2.7     0.0     0.7   !            !    205   !
+------------------------------------------+           +----------------------------------------+            +----------+

+---------- File Cache Attempt Rate (per second) ----------+     +---------------- File cache Effectivness ----------------+
!                                                          !     !                                                         !
!   Dir     Dir             File    File             Bit   !     !   Dir     Dir             File    File             Bit  !
!   FCB     Data   Quota    Id      Hdr    Extent    Map   !     !   FCB     Data   Quota    Id      Hdr    Extent    Map  !
!  ------  ------  ------  ------  ------  ------  ------  !     !  ------  ------  ------  ------  ------  ------  ------ !
!    0.9     1.4     0.0     0.1     2.3     1.4     0.2   !     !  97.7%   85.7%    0.0%   99.6%   90.3%   98.0%    5.0% !
+----------------------------------------------------------+     +---------------------------------------------------------+

+------------------------------------ Disk Statistics -----------------------------------------+
!                              Work                                             Serv    Resp                     !
!                              Avail   Paging  Swping  Contlr   Rate    Read   Remote  Time    Time   Queue  Space   !
!                               %        %       %       %      (/s)     %     I/O%    (ms)    (ms)  Length  Used %  !
!                              ------  ------  ------  ------  ------  ------  ------  ------  ------ ------  ------  !
! $255$DBA1                     0.0     0.0     0.0     0.0     0.0     0.0     0.0      0       0     0.0    35.3   !
! $255$DJA9                    10.0     0.0     0.0    15.7     3.1    84.7     8.0     32      33     0.1    84.8   !
```

Figure 2–7 VAX SPM Tabular Report for a 30-Minute Period

Table 2–1 Tabular Report Fields to Diagnose a Limiting Resource

Resource	Tabular Report Field	Explanation
Memory	Free Pages	Size of the free page list
	Total MEMutl	The percentage of available memory that is used
	Mem (under AVE Mem)/CPU Queues	The average number of processes in the memory queue
	InSWP	The number of process inswaps performed
	Swaper CPU%	The percentage of CPU time used by the SWAPPER process. This includes time for swapping, modified page writing, and process working set trimming activities
	Page Faults	The total number of page faults per second (both hard and soft) including system faults
	System Faults	The total number of system page faults
I/O	Direct I/Os	The number of direct I/O operations performed per second for all disks in the system (this includes RMS I/O and excludes page and swap I/O and I/O to mapped image sections)
	Buffrd I/Os	The number of buffered I/O operations performed per second
	Rate(/s)-under Disk Statistics	The number of I/O operations per second for the device
CPU	CPU (under AVE Mem/CPU Queues)	The average number of processes in the CPU queue
	Total Idle	The percentage of time that the CPU was idle. In a multiprocessor system, statistics for additional processors are given below the data for the primary processor

For example, a value of Total MEMutl greater than 90% indicates a possible memory limitation. Likewise, a value of Rate(/s) greater than 30 for any disk indicates a possible I/O limitation. A value of Total Idle less than 15% indicates a possible CPU limitation.

Chapter 7 provides more information about the tabular report. Descriptions of the tabular report fields are in the *VAX SPM Reference Manual.*

2.6 Generating the System Load Balance Display

The Load Balance display provides an overview of current system performance. Use the Load Balance display as a starting point for system tuning or for on-line monitoring of system performance or resource utilization.

To run the VAX SPM video displays on a VT100 terminal that supports DEC_CRT characteristics, set the terminal characteristics to DEC_CRT by typing:

```
$ SET TERMINAL/DEC_CRT
```

Type the following command to display the VAX SPM Load Balance display:

```
$ PERFORMANCE DISPLAY=INVESTIGATE
```

You can exit from the video display screens at any time by entering CTRL/Z or by typing the word EXIT, then pressing RETURN.

If you are working at an ANSI display terminal such as a VT100, the VAX SPM Load Balance display resembles the one shown in Figure 2–8.

The ANSI version of the Load Balance display is a bar graph of eight performance indicators. The indicators in the top half are considered "good" as they increase in numerical value, and the indicators in the bottom half are considered "bad" as they increase. Because of these conventions, a well-balanced system is represented when bars appear mostly in the upper half of the diagram.

If you are working at a ReGIS-compatible terminal such as a VT340, the System Overview display is the default. Chapter 12 describes the System Overview display.

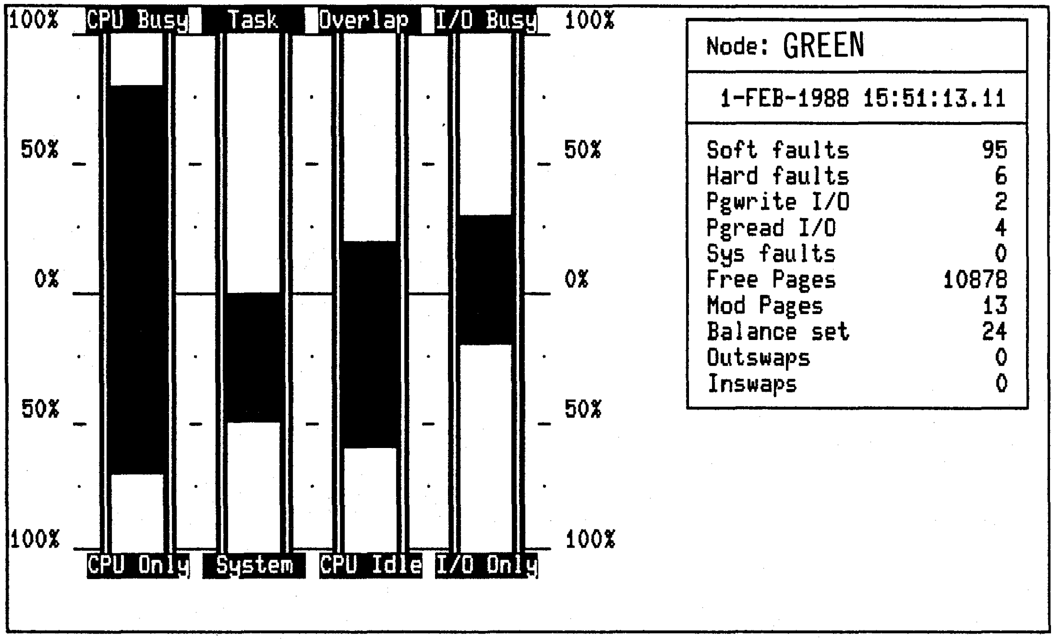
To view the Load Balance display, you must specifically request the Load Balance display by entering the following command:

```
DISPLAY LOAD
```

When you want to display the Load Balance display from the System Overview display, type DISPLAY LOAD. The SPM> prompt appears and the current display pauses while your command is entered.

The ReGIS version of the Load Balance display is the Kiviat display (Figure 2–9). It is a graph of eight indicators plotted in a circular fashion. These indicators are plotted on the radii of a circular diagram where each radius is evenly spaced around the circle. Each point is a system metric related to CPU activity, disk I/O activity, or both. The center of the circle represents 0% for a given metric; the intersection of the circle and the outer radius represents 100% for a given metric.

Figure 2–8 ANSI Version of System Load Balance Video Display



```
Node: GREEN

 1-FEB-1988 15:51:13.11

Soft faults          95
Hard faults           6
Pgwrite I/O           2
Pgread I/O            4
Sys faults            0
Free Pages        10878
Mod Pages            13
Balance set          24
Outswaps              0
Inswaps               0
```

Half the indicators are considered "good" when they increase in numerical value, while the other half are considered "bad" when they increase in value. The "good" and "bad" indicators are plotted alternately. The "good" indicators are CPU Busy, Overlap, I/O Busy, and Task CPU. The "bad" indicators are CPU Only, I/O Only, CPU Idle, and System CPU.

Because of these conventions, the more a system is improved or tuned, the more closely its load balance approaches as a "star" shape.

2.7 Chapter Summary

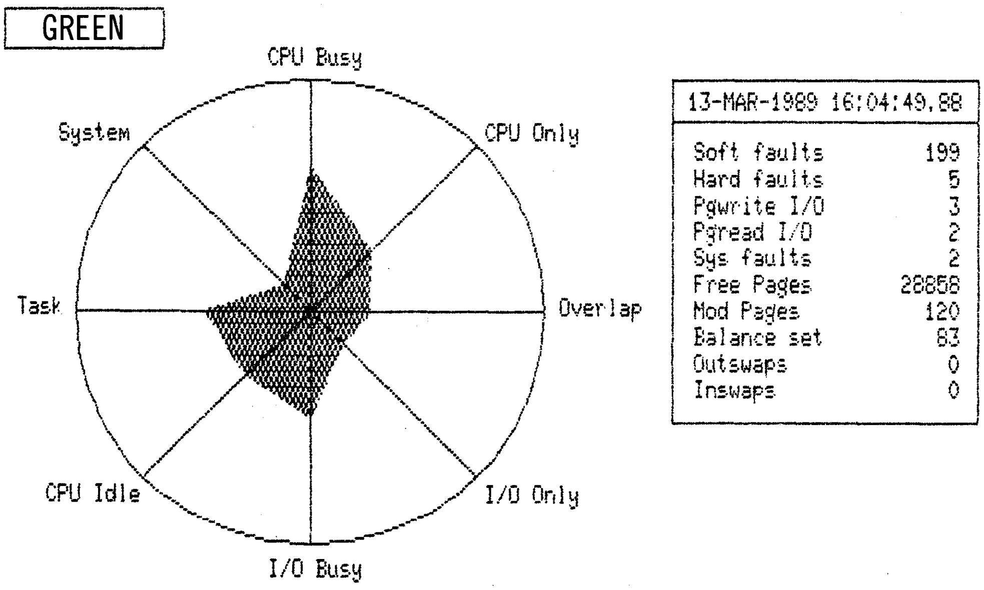
This chapter described some basic VAX SPM activities. The activities and their commands follow.

- Determining if Collections are Running

  Only one SPM_TUNE and one SPM_CAPACITY process can run simultaneously. Before starting a collection process, use the following commands to determine if a tune or capacity process is running:

  ```
  $ PERFORMANCE COLLECT=TUNE/INQUIRE
  $ PERFORMANCE COLLECT=CAPACITY/INQUIRE
  ```

Figure 2–9 ReGIS Version of System Load Balance Video Display



```
13-MAR-1989 16:04:49.88

Soft faults          199
Hard faults            5
Pgwrite I/O            3
Pgread I/O             2
Sys faults             2
Free Pages         28858
Mod Pages            120
Balance set           83
Outswaps               0
Inswaps                0
```

When a collection process is not running, the following message displays:

```
%SPM-E-CONORU, collections not running
```

If a tune or capacity collection process is running, a status report displays. If both tune and capacity collection processes are running, wait until one process stops before starting collections.

- Starting and Stopping Collections

  If a tune or capacity collection process is not running, start the collection process by entering one of the following commands:

```
$ PERFORMANCE COLLECT=TUNE
$ PERFORMANCE COLLECT=CAPACITY
```

  Stop a tune or capacity collection process by using the following commands:

```
$ PERFORMANCE COLLECT=TUNE/STOP
$ PERFORMANCE COLLECT=CAPACITY/STOP
```

- Generating a System Summary Graph

  A system summary graph provides an overview of system activity and is a good initial report when evaluating system performance.

  Use the following command to generate a system summary graph from a log file:

```
$ PERFORMANCE REPORT=LOG_FILE/NOTABULAR/GRAPH=SUMMARY SPM$COLLLECT_TUNE
```

- Generating a Tabular Report

  A tabular report provides detailed performance information and contains most of the information you need to diagnose and isolate a limiting resource.

  Use the following command to generate a tabular report:

  ```
  $ PERFORMANCE REPORT=LOG_FILE/TABULAR/NOGRAPH SPM$COLLECT.DAT
  ```

- Generating the Investigate Load Balance Video Display

  The Load Balance display provides an overview of current system performance. Use the Load Balance display as a starting point for system tuning or for on-line monitoring of system performance or resource utilization.

  On a VT100 terminal that supports DEC_CRT characteristics, set these characteristics with the following command:

  ```
  $ SET TERMINAL/DEC_CRT
  ```

  Type the following command to display the VAX SPM Load Balance display:

  ```
  $ PERFORMANCE DISPLAY=INVESTIGATE
  ```

  For a ReGIS-compatible terminal such as a VT340, the System Overview display precedes the Load Balance display. Type the following command to show the Load Balance display:

  ```
  SPM> DISPLAY LOAD
  ```

  Now that you have completed this chapter and have become familiar with some basic VAX SPM activities, you may want to begin collecting performance data as described in Chapter 3.

3 Collecting Performance Data

This chapter describes how to:

- Use the PERFORMANCE collection commands for conventional data collections.
- Use the SPM$MANAGER.COM command procedure for continuous automatic data collections.

3.1 Conventional Data Collection

3.1.1 Overview

SPM provides two PERFORMANCE COLLECT commands:

```
$ PERFORMANCE COLLECT=TUNE
```

```
$ PERFORMANCE COLLECT=CAPACITY
```

To begin collecting performance data, your account requires either the SETPRV privilege or all the following privileges:

```
ALTPRI    CMKRNL    DETACH    PRMMBX    PSWAPM    NETMBX
SYSLCK    SYSNAM    SYSPRV    TMPMBX    WORLD
```

The two PERFORMANCE COLLECT commands are capable of collecting the same types of data. Although these commands may be used interchangeably for collecting data, COLLECT=TUNE is often reserved for performance evaluation and COLLECT=CAPACITY is often reserved for historical reporting.

Each command creates a detached process that performs the data collections. The names of these detached processes are SPM_TUNE and SPM_CAPACITY, respectively. Because data is collected by a detached process, you are free to give other commands at your terminal after starting the process. You can also log off the system and the detached process will continue to run. Table 3–1 lists all classes of data collected by the Tune and Capacity collection processes and the types of statistics provided for each class.

Although the examples in this chapter use the PERFORMANCE COLLECT=TUNE command, the PERFORMANCE COLLECT=CAPACITY command can be used, also.

Table 3–1 Classes of Data Provided by SPM

Class	Types of Statistics Provided
Default Data	
CPU	CPU Identification, Total Idle, Busy Wait, Interrupt Stack, Kernel, Executive, Supervisor, User, Compatibility, System, Task per CPU, Lost CPU, and CPU Overlap
Disk	Work Available, Paging Percent, Swapping Percent, Controller Percent, Rate/Second, Service Time (in milliseconds), Response Time, Queue Length, and Percentage of space used. Server data is also collected, including Percentage of Work Available, Paging and Swapping, and Queue Length
I/O	Buffered and Direct I/O Rates per Second, Logical Name Translations, Mailbox Reads and Writes, and File I/O Rates per Second
Memory	Total Memory Utilization (including Paged, User, and Modified Memory Utilized), Average Memory Queue, and Swapper Counts
Page Fault	Paging Rates/Second (including System Page Faults, Pages Read and Written, Read and Write I/Os, Free List, Modified List, Bad List, Demand Zero Faults, Global valid Faults, Transition State Faults, and Write in Progress Faults), and Hard and Soft Faults
XQP Cache	File Cache Attempt Rates per Second and File Cache Effectiveness for the Direct File Control Processor (FCB), Direct Data, Quota, File ID, File Header, Extent, and Bit Map Caches
Optional Data	
Device	I/O Rates
File Primitives	Number of Operations, Disk Reads and Writes, Cache Hits, Hit Ratio, CPU Time in Seconds and Page Faults for: Access, Create, Deaccess, Delete, Modify, ACP Controller, Lookup, Enter, Allocate, and Attributes operations
[1]Global Sections	Name, Version Number, Status (for example, Writable, Permanent or Temporary, System- or Group-wide), Page Count, and Number of Page Table Entries
[1]Hardware Configuration	System Identification, Node Name, Processor Type, Slot, Adapter, Controller, Unit, Class, Type, and Size
[1]Installed Images	Name of Image, How Installed, Entry Access Count, Current and Maximum Count of Shared Accesses, Privileges, and Global Section Count
Lock	Lock Rates per Second (including Local, Incoming, or Outgoing), Number of New Locks Requested, Not Granted, Deadlocks Searched and Found, Total Locks, and Total Resources that can be locked

[1] This data is collected only once at the start of collections.

Table 3–1 (Cont.) Classes of Data Provided by SPM

Class	Types of Statistics Provided
Process	Process Name, UIC, Priority, State, Image Count, CPU Time (in minutes), Direct I/O Rate/Second, Buffered I/O Rate/Second, Page Fault and Fault I/O Rates, and Minimum, Average, and Maximum Size of Working Sets
Scheduler	Scheduler States Organized by Time-sharing and Real-Time Priorities
[1]System Parameters	Parameter Name and Current Value
System Communication	Datagrams, Block Transfers, and Sequenced Messages (including Number Sent, Received, Discarded, or Queued for Nodes on a VAXcluster System Present During the Reporting Interval)

[1]This data is collected only once at the start of collections.

The COLLECT commands take one parameter: the node name parameter. If this parameter is omitted, the current node is assumed.

The node name parameter specifies the node on which collections are invoked. Collections may be invoked on any node in the VAXcluster system or on remote stand-alone nodes. Chapter 21 provides additional information on collecting performance data on a VAXcluster system and Chapter 22 describes how to invoke collections on remote nodes.

In addition to specifying the node on which collections take place, the presence or absence of the node name parameter in the command line determines the directory in which the data file is created. In the following example, the node name is omitted from the command line:

```
$ PERFORMANCE COLLECT=TUNE
```

The node from which the command was typed is assumed and the data file named SPM$COLLECT_TUNE.DAT is created in the current default directory.

When the node name is specified, the data file SPM$COLLECT_TUNE_nodename.DAT is created in the directory of the account from which SPM$MANAGER.COM is run for that node. To collect on node GREEN, give the following command:

```
$ PERFORMANCE COLLECT=TUNE GREEN
```

The above command creates a data file named SPM$COLLECT_TUNE_GREEN.DAT in the SPM account on node GREEN. This command also assumes that node GREEN is part of the local cluster.

To specify all nodes in a VAXcluster system, type an asterisk (*) in place of the node name as shown in the following command:

```
$ PERFORMANCE COLLECT=TUNE *
```

The above command creates a data file named SPM$COLLECT_TUNE_nodename.DAT in the directory of the SPM account for each node in the cluster.

The SPM collection commands have many qualifiers. These qualifiers allow you to create a collection process that suits your needs. This chapter discusses only those qualifiers needed to get started collecting performance data. See the *SPM Reference Manual* for a full description of all collection command qualifiers.

The previous examples specify default values for all the command qualifiers. Table 3–2 lists some collection qualifiers and the effects of their defaults on the collection process.

Table 3–2 Default Values of Collection Qualifiers

Collection Qualifier	Default Action
/BEGINNING[=time]	Collection process begins when you type the command line.
/ENDING[=time]	Collection process continues until you stop it.
/INTERVAL[=seconds]	Elapsed time between data collection events (collection interval) is 300 seconds.
/CLASS=(item[,...])	Statistics for CPU, DISK, I/O, MEMORY, PAGE_FAULT, and XQP_CACHE are collected.
/OUTPUT[=file]	Creates a data file named SPM$COLLECT_TUNE.DAT in the default directory. If a node name was specified, the log file SPM$COLLECT_TUNE_nodename.DAT is created on the node in the directory specified for the SPM account during installation.

3.1.2 Stopping and Starting a Collection Process

To start a collection process on the current node, type the following command:

```
$ PERFORMANCE COLLECT=TUNE
```

To stop a collection process at any time, type the following command:

```
$ PERFORMANCE COLLECT=TUNE/STOP
```

Use the /BEGINNING and /ENDING qualifiers, to specify a beginning an ending time for your collection process. Times for the /BEGINNING and /ENDING qualifiers can be given in absolute or delta formats. See the *VMS DCL Concepts Manual* for valid time formats.

With the /BEGINNING and /ENDING qualifiers, you are able to type the collection command and at the same time specify when to begin the collection and when to end the collection of data. The following example specifies absolute times for the /BEGINNING and /ENDING qualifiers.

```
$ PERFORMANCE COLLECT=TUNE/BEGINNING=04-FEB-1988:07:00-
  /ENDING=04-FEB-1988:17:00
```

The collection process begins at 7 a.m. on February 4, 1988, and ends at p.m. on that same day.

The next example specifies delta times for the /BEGINNING and /ENDING qualifiers.

```
$ PERFORMANCE COLLECT=TUNE/BEGINNING="+00:10:00"/ENDING="+02:10:00"
```

This command creates the tune collection process, directs it to hibernate for 10 minutes, collect data for 2 hours, and then terminate.

3.1.3 Specifying the Sampling Interval

Use the /INTERVAL qualifier to specify the frequency for collecting and writing data to the log file. A sample is taken when the collection process begins, and thereafter data is collected and written to the log file at each sampling interval.

The sampling interval must be in the range of 1 to 3600 seconds. A smaller interval captures rapid fluctuations in the values of performance metrics at the expense of larger log files and use of more system resources by SPM. The default interval when the /INTERVAL qualifier is omitted is 300 seconds (5 minutes). Each time data is collected and written, the value of the current sample in the SPM Collection Process Status report (Figure 2–2) is incremented by 1.

A sampling interval of 300 seconds is recommended for collecting system tuning data, and a sampling interval of 900 seconds or more is recommended for collecting historical data. The following command specifies a sampling interval of 120 seconds.

```
$ PERFORMANCE COLLECT=TUNE/INTERVAL=120
```

3.1.4 Specifying Classes of Data for Collection

Use the /CLASS qualifier and keywords to specify the classes of data for which you want to collect statistics. Table 3–3 lists all keywords for the /CLASS qualifier and their meanings. Examples of the types of statistics provided for each class are shown in Table 3–1.

Table 3–3 Classes of Data Collected by SPM

/CLASS Keyword	Data Collected
ALL	All statistics
ANALYZE	CPU, Hardware Configuration, Process, Device, I/O, Scheduler, Disk, Memory, System Parameters, File Primitives, Page Fault, and XQP Cache
CPU	**CPU statistics**
DEFAULT_STATISTICS	**CPU, DISK, I/O, MEMORY, PAGE_FAULT, and XQP_CACHE statistics**
DEVICE	Device rate/second statistics for devices specified by /DEVICE

Classes collected by default are distinguished by bold type.

Table 3–3 (Cont.) Classes of Data Collected by SPM

/CLASS Keyword	Data Collected
DISK	**Disk statistics for disks specified by /DISK and server statistics**
FILE_PRIMITIVES	File system primitive operations statistics (for example, file system read/writes, the number of times they occur, and the amount of CPU and page faulting they incur)
GLOBAL_SECTIONS	Global section data
HARDWARE_CONFIGURATION	Hardware configuration information
INSTALLED_IMAGES	Installed image information
IO	**I/O statistics**
LOCK	Lock statistics
MEMORY	**Memory statistics**
PAGE_FAULT	**Page fault statistics**
PROCESS[=([NO]EXTENDED, [NO]IMAGE,ALL)]	Process statistics
SCHEDULER	Scheduler statistics
SYSGEN_PARAMETERS	System parameter information
SYSTEM_COMMUNICATION	System Communication Services (SCS) data—communication data between nodes in a cluster
XQP_CACHE	**XQP caching statistics**

Classes collected by default are distinguished by bold type.

Notes on the /CLASS Qualifier Keywords

- **The ALL Keyword**—Use the /CLASS=ALL qualifier and keyword to collect all classes of data. Collecting all classes is a good approach for the beginning SPM user. After becoming familiar with the system and your collection needs, you can eliminate classes you find unnecessary.
- **Negated Keywords**—Use the negated form of keywords to define classes you do not want collected when using the DEFAULT_STATISTICS and ALL keywords. The following command causes all classes of data to be collected except global sections and hardware statistics:

```
$ PERFORMANCE COLLECT=TUNE/CLASS= -
  (ALL,NOGLOBAL_SECTIONS,NOHARDWARE_CONFIGURATION)
```

- **The DEFAULT_STATISTICS Keyword**—Default statistics are always collected unless specifically negated.

  In the following command, only the LOCK and SCHEDULER keywords are specified; however, the default statistics CPU, DISK, I/O, MEMORY, PAGE_FAULT, AND XQP_CACHE are also collected:

```
$ PERFORMANCE COLLECT=TUNE/CLASS=(LOCK,SCHEDULER)
```

To collect only lock and scheduler statistics, negate the DEFAULT_STATISTICS keyword as shown in the following command:

```
$ PERFORMANCE COLLECT=TUNE/CLASS=(NODEFAULT_STATISTICS,LOCK,SCHEDULER)
```

Use negated keywords to collect most default statistics as shown in the following command:

```
$ PERFORMANCE COLLECT=TUNE/CLASS=(NOXQP_CACHE,NOPAGE_FAULT)
```

The default statistics CPU, DISK, I/O, and MEMORY are collected. Because the keywords XQP_CACHE and PAGE_FAULT are negated, statistics for these classes are not collected.

- **The PROCESS keyword**—Extended process metrics reports in which the image name is part of the process identification are helpful in system tuning. In order to generate these reports, you must have collected extended process metric and image data. Use the following command to specify all process data, including process metrics, extended process metrics, and image data:

  ```
  $ PERFORMANCE COLLECT=TUNE/CLASS=(PROCESS=ALL)
  ```

  Note that selecting /CLASS=ALL also ensures that extended process metrics and image data is collected.

3.1.5 Specifying Disks and Devices for Collection

Use the /DEVICE and /DISK qualifiers to specify the devices and disks for which you want performance statistics. SPM recognizes a "$ SHOW DEVICE" style naming convention for disk and device specification. For example, device names specified for collection may be in the form of physical device names, logical names, or abbreviated device names. Disk and device names specified for collection may contain leading underscores or trailing colons. If a disk or device specified on the command line is not available for data collection, a warning message is issued and the data collection continues.

When an abbreviated disk or device name is supplied, one of three actions is taken:

- If the disk or device name is abbreviated (for example, DU or XE), then data collection is initiated for all disks or devices beginning with what was specified. Other examples of abbeviated device names are GREEN$, YELLOW$, and GREEN$DU, where GREEN and YELLOW are node names in a VAXcluster system.
- If the controller designation is omitted (for example, DB2 or XE1), then data collection is initiated for all disks or devices on all controllers with the specified unit number (for example, DBA2 and DBB2).
- If the unit number is omitted (for example, DBA or XEA), then data collection is initiated for all disks or devices on the specified controller (for example, DBA1 and DBA2).

The following command collects disk statistics for specific devices only. It collects disk statistics for disks DRA0 and DRA1.

```
$ PERFORMANCE COLLECT=TUNE/DISK=(DRA0,DRA1)/OUTPUT=SPM$COLLECT_TUNE.DAT
```

Note that it is not necessary to specify /CLASS=(DISK), because disk statistics are collected by default. The /DISK qualifier specifies the disks for which collections are initiated.

In the following command, optional operation count rate/second data for disk DBA1 and devices LPA0 and TTB6 is collected.

```
$ PERFORMANCE COLLECT=TUNE/CLASS=(DEVICE)/OUTPUT=LDEV.DAT-
  /DEVICE=(DBA1,LPA0,TTB6)
```

Note that it is necessary to specify /CLASS=(DEVICE) because device statistics are not collected by default. Note also that the /DEVICE and /DISK qualifiers collect different kinds of data but only the /DEVICE qualifier can specify devices other than disk.

3.1.6 Specifying an Output Log File

As data is collected, it is written to a log file. Use the /OUTPUT qualifier to specify the name of a log file. If you do not specify a log file or a node name, the log file SPM$COLLECT_TUNE.DAT is created in the current default directory.

The following command specifies the log file name COLLECT.DAT created in the default directory for the current node.

```
$ PERFORMANCE COLLECT=TUNE/OUTPUT=COLLECT.DAT
```

If you do not specify a log file name but do specify a node name, the log file SPM$COLLECT_TUNE_nodename.DAT is created on the specified node in the account defined as the SPM account during installation.

3.2 Automatic Data Collection

The command procedure SPM$MANAGER.COM provided during SPM installation automates the collection and reporting process in the followinę ways:

- Stops and starts performance collections each day
- Evaluates data and sends a VAXmail message as notification wheneve performance thresholds are exceeded
- Allows users to customize collection, evaluating, and reporting capabilities to meet their requirements

Before running the SPM$MANAGER.COM command procedure, you mus set up the environment in which it is to run. There are two ways you can set up this environment: automatically using the SPM$MANAGER_CONFIGURE.COM command procedure described in Section 3.2.1, or by performing the instructions in Section 3.2.2.

You can use the SPM$MANAGER.COM command procedure to run the SPM$COLLECT_TUNE and SPM$COLLECT_CAPACITY processes simultaneously. Refer to Section 3.2.3 for instructions.

Section 3.2.4 describes SPM$MANAGER.COM error reporting.

3.2.1 Customizing SPM$MANAGER.COM Using SPM$MANAGER_CONFIGURE.COM

The SPM$MANAGER_CONFIGURE.COM command procedure is an interactive command procedure that sets up the environment for running SPM$MANAGER.COM and creates the SPM$MANAGER_SYMBOLS.COM command file.

The SPM$MANAGER_CONFIGURE.COM performs the following two tasks related to the symbol values in the SPM$MANAGER_SYMBOLS.COM file:

- Displays each symbol, its description, and default value. You can either accept the default value or type in another value. The default values are appropriate for most collection and reporting requirements
- Checks the validity of symbol definition values

Follow the instructions below to set up the environment and run the SPM$MANAGER.COM command procedure:

1. Log in to the account created for SPM during installation (usually SPM) or to an account with privileges to access the SPM account.
2. Type the following command to invoke the SPM$MANAGER_CONFIGURE.COM command procedure:

   ```
   $ @SPM$EXAMPLES:SPM$MANAGER_CONFIGURE
   ```

3. Respond to each prompt by either pressing RETURN to accept the default value or by supplying a value followed by RETURN.

   Although values for all the symbols can be changed to suit your specific collection and reporting needs, the numbered symbols in Example 3–1 are ones you may want to change to begin running the SPM$MANAGER.COM command procedure.

4. When you have changed the values or accepted the defaults for all symbols, the command procedure writes the SPM$MANAGER_SYMBOLS.COM file and terminates.
5. To ensure that SPM$MANAGER.COM runs continuously, include the following commands in your site-specific start-up file:

   ```
   $! Load the SPMTIMER device driver
   $@SYS$STARTUP:SPM$STARTUP
   $! Establish the node-specific collection initialization file, if
   $! desired.  This is not required to run SPM.
   $NODE=F$GETSYI("NODENAME")
   $ SUBMIT /USER={value_of_spm_username}/NOPRINT -
           /PARAMETER=("BOOT",{"CAPACITY" or "TUNE"}) -
           /QUEUE='NODE'{node_specific_batch_queue_extension} -
           {value_of_procedure_directory}SPM$MANAGER -
           /LOG={value_of_batchlog_directory}'NODE'_SPM$MANAGER.LOG
   ```

Because the purpose of the SUBMIT command is to start collections on the node that is booting, it is important that the batch job be submitted to the queue of the node that is booting and no other node in the cluster. In the above example, this is accomplished by a convention that uses the node name as the queue-name prefix and a queue name common to all nodes as the queue-name extension. The use of this convention for nodes RED, BLUE, and GREEN would create queue names of RED$SYS_BATCH, BLUE$SYS_BATCH, and GREEN$SYS_BATCH, respectively.

Thereafter, whenever a node is rebooted, the SPM$MANAGER.COM command procedure is submitted to the batch queue on that node by the node's start-up file.

6 Submit SPM$MANAGER.COM to the batch queue by typing the following command:

```
$ SUBMIT /USER={value_of_spm_username}/NOPRINT/PARAMETER=("BOOT", -
        {"CAPACITY" or "TUNE"})
        /QUE='NODE'{node_specific_batch_queue_extension} -
        {value_of_procedure_directory}SPM$MANAGER -
        /LOG={value_of_batchlog_directory}'NODE'_SPM$MANAGER.LOG
```

For example, to start a tune collection for a node where the following is true:

Node name is GREEN
SPM account is named SPM
Default directory is [SPM]
Procedure subdirectory in the SPM account is named [.PROCEDURE]
Batch subdirectory in the SPM account is named [.BATCHLOG]

Use following command:

```
$ SUBMIT /USER=SPM/NOPRINT/PARAMETER=("BOOT","TUNE") -
        /QUE=GREEN$SYS_BATCH -
        [SPM.PROCEDURE]SPM$MANAGER -
        /LOG=[SPM.BATCHLOG]GREEN_SPM$MANAGER.LOG
```

The command must be repeated on each node on a VAXcluster system. The command procedure SPM$MANAGER.COM stops and starts collections automatically each day for all nodes on a VAXcluster system.

3.2.2 Customizing SPM$MANAGER.COM

Perform these instructions to set up the SPM$MANAGER.COM command procedure and start automatic data collections on your system without using the SPM$MANAGER_CONFIGURE.COM command procedure.

1 Log in to the account created for SPM during installation (usually SPM).

2 Create the directories for the SPM$MANAGER files. The following directories are recommended based upon SPM$MANAGER.COM symbols shown in Example 3–1.

[.BATCHLOG]	The batch log files
[.PROCEDURE]	The SPM$MANAGER procedure
[.LOGFILE]	SPM data files
[.HISTORY]	SPM history file

3 Copy SPM$MANAGER.COM from the [.SPM] subdirectory of the SYS$EXAMPLES account (SYS$COMMON:[SYSHLP.EXAMPLES.SPM]) to the procedure subdirectory [.PROCEDURE].

4 Create the file SPM$MANAGER_SYMBOLS.COM in the top level directory of the SPM account or the account from which SPM$MANAGER.COM is to run. This file will contain symbols for the SPM$MANAGER.COM command procedure. Edit the SPM$MANAGER.COM file, and locate the symbols and their definitions. The symbols and definitions begin and end with a single row of asterisks. Extract the symbols and definitions into the symbol file (SPM$MANAGER_SYMBOLS.COM). Be sure to delete the exclamation points in your symbol file.

5 Create a history file called SPM$HISTORY.DAT in the [.HISTORY] subdirectory to accommodate data for historical reporting purposes. (Note: Do not do this if it already exists!)

```
$PERFORMANCE ARCHIVE=HISTORY/CREATE SPM$HISTORY.DAT
```

The above command creates an empty 3000-block history file with a 15-minute collection interval.

If you already have a history file, move it to the [.HISTORY] subdirectory of the SPM account so that log files can be archived to it after they are collected.

6 Edit the SPM$MANAGER_SYMBOLS.COM file and change the values of symbols as necessary to customize the command file to your system.

Example 3–1 shows the SPM$MANAGER_SYMBOLS.COM file. Although values for all the symbols can be changed to suit your specific collection and reporting needs, the numbered symbols are ones you may wish to change before running the command procedure. The corresponding numbers below describe the values for these symbols, the effects of these values and instructions for changing them.

Example 3–1 SPM$MANAGER_SYMBOLS.COM

```
$❶    SPM_USERNAME     =="SPM"
$❷    CLUSTER_NODES    == ""
$❸    BEGIN_COLLECTION == "07:55"
$❹    END_COLLECTION   == "17:05"
$❺    COLLECTION_INTERVAL == "300"
$❻    COLLECTION_QUALIFIERS == "/CLASS=ALL/DEVI=(LP,XE,MU,MT,MS)"+-
                            "/INTERVAL=''COLLECTION_INTERVAL'"
$    LOGFILE_DIRECTORY == "[.LOGFILE]"
$    BATCHLOG_DIRECTORY == "[.BATCHLOG]"
$    PROCEDURES_DIRECTORY == "[.PROCEDURE]"
$    JOB_NAME         == "SPM$MANAGER"
$    REPORT_TYPE == "/GENERAL"
$    LOG_FILE DESTINATION == ""
$    BATCH_QUEUE == "SYS$BATCH"
$    NODE_GRAPH_QUALIFIER == "/GRAPH=(PAGE_FAULTS:100,SWAP_COUNT:30,"+-
           "MEMORY:85,DIRECT_IO:50,BUFFERED_IO:50,D_RATE:20,"+-
            "D_RESPONSE:60,CPU:85,LOST_TIME:5)"
$    CLUSTER_GRAPH_QUALIFIER == "/GRAPH=(MEMORY:85, D_RATE:20,"+-
             "D_RESPONSE:60,CPU:85)"
$    CLUSTER_NAME == ""
$    SELECT_QUALIFIER == "/SELECT=GREATER_THAN:50"
$    SHIFT_QUALIFIER  == "/SHIFT=PRIME"
$    P_HOURS_QUALIFIER == "/P_HOURS=(8-16)"
$    P_DAYS_QUALIFIER == "/P_DAYS=(NOSUNDAY,MONDAY,TUESDAY,WEDNESDAY,"+-
                        "THURSDAY,FRIDAY,NOSATURDAY)"
$❼   DISTRIBUTION_LIST == "SYSTEM"
$   HISTORY_FILE_DIRECTORY == "[.HISTORY]"
$   HISTORY_FILE == "SPM$HISTORY.DAT"
$   ARCHIVE_QUALIFIERS == "/CLASS=(ALL,NOFILE_PRIMITIVES,"+-
                       "NOSCHEDULER,NOXQP_CACHE)"
```

❶ The user name assigned to the SPM account during installation. If a name other than SPM was specified for the SPM account during installation, type that name here.

❷ The name of a text file containing VAXcluster system node names or an asterisk (*) to specify all nodes in a VAXcluster system. If you are running SPM on a VAXcluster system, create a text file in the top level directory of the account designated for SPM. In this file, type the node names of the cluster members on which you wish to run the SPM$MANAGER.COM command procedure, or type an asterisk (*) to specify all cluster members. Node names should be typed one per line with no leading or trailing characters. Supply the name of this text file as the value of the CLUSTER_NODES symbol. If no value is provided for this symbol, a single-node system is assumed.

❸ The time collections start each day. If you do not want collections to begin at 7:55 a.m., type in the time you want collections to begin.

❹ The time collections stop each day. If you do not want collections to end at 5:05 p.m., type in the time you want collections to end.

Example 3–1 Cont'd. on next page

Example 3–1 (Cont.) SPM$MANAGER_SYMBOLS.COM

❺ The frequency at which data is collected. If you do not want data to be collected at an interval of 300 seconds (5 minutes), type in the interval you want. Interval values can be from 1 to 3600 seconds. For performance collections, a typical interval is 300 seconds. For historical reporting purposes, a typical interval is 900 seconds or more.

❻ The classes of data collected. If you do not want all classes of data collected, specify the classes you want. If you do not want data collected for the devices LP, XE, MU, MT, and MS, modify values for the /DEVICE qualifier to correspond to devices on your system for which you wish to collect data. Use the DCL command SHOW DEVICE for a list of the devices on your system.

❼ The account to receive notification by VAXmail if threshold values are exceeded. To notify accounts in addition to the "SYSTEM" account, type the names of the other accounts separated by commas. If no account names are entered, generation and mailing of graph reports will not occur.

7 Define the logical SPM$MANAGER_SYMBOLS to point to the symbol file by placing the following command in the LOGIN.COM file of the "SPM" account.

```
$ DEFINE SPM$MANAGER_SYMBOLS SPM$MANAGER_SYMBOLS.COM
```

8 To ensure that SPM$MANAGER.COM runs continuously, include the following commands in your site-specific start-up file:

```
$! Load the SPMTIMER device driver
$@SYS$STARTUP:SPM$STARTUP
$! Establish the node-specific collection initialization file, if
$! desired.  This is not required to run SPM.
$NODE=F$GETSYI("NODENAME")
$ SUBMIT /USER={value_of_spm_username}/NOPRINT/PARAMETER=("BOOT", -
  {"TUNE" or "CAPACITY"}) -
        /QUEUE='NODE'{node_specific_batch_queue_extension} -
        {value_of_procedure_directory}SPM$MANAGER -
        /LOG={value_of_batchlog_directory}'NODE'_SPM$MANAGER.LOG
```

Because the purpose of the SUBMIT command is to start collections on the node that is booting, it is important that the batch job be submitted to the queue of the node that is booting and no other node in the cluster. In the above example, this is accomplished by a queue naming convention that uses the node name as the queue-name prefix and a queue name common to all nodes as the queue-name extension. The use of this convention for nodes RED, BLUE, and GREEN creates the queue names RED$SYS_BATCH, BLUE$SYS_BATCH, and GREEN$SYS_BATCH, respectively.

Thereafter, whenever a node is rebooted, the SPM$MANAGER.COM command procedure is submitted to that node's batch queue by the system-specific start-up file.

9 Submit SPM$MANAGER.COM to the batch queue by typing the following command:

```
$ SUBMIT /USER={value_of_spm_username}/NOPRINT/PARAMETER=("BOOT", -
        {"CAPACITY" or "TUNE"}) -
        /QUE='NODE'{node_specific_batch_queue_extension} -
        {value_of_procedure_directory}SPM$MANAGER -
        /LOG={value_of_batchlog_directory}'NODE'_SPM$MANAGER.LOG
```

For example, to start a tune collection for a node where the following is true:

Node name is GREEN
SPM account is named SPM
Procedure subdirectory in the SPM account is [.PROCEDURE].
Batch subdirectory in the SPM account is named [.BATCHLOG].

Use following command:

```
$ SUBMIT /USER=SPM/NOPRINT/PARAMETER=("BOOT","TUNE") -
        /QUE=GREEN$SYS_BATCH
        [.PROCEDURE]SPM$MANAGER -
        /LOG=[.BATCHLOG]GREEN_SPM$MANAGER.LOG
```

The command must be repeated on each node on a VAXcluster system. The command procedure SPM$MANAGER.COM stops and starts collections automatically each day for all nodes on a VAXcluster system.

3.2.3 Running Two Collection Processes Simultaneously

Follow these instructions to set up the SPM$MANAGER.COM command procedure to run the SPM_TUNE and the SPM_CAPACITY processes simultaneously:

1 Select either the method described in either Section 3.2.1 or Section 3.2.2 and create a symbols file for the SPM_TUNE process.

2 Specify SPM_TUNE as the value for the JOB_NAME symbol and a collection interval appropriate for tuning for the COLLECTION_INTERVAL symbol. A value of 300 seconds is usually used for performance tuning collections.

3 If you are creating the symbols file using the instructions in Section 3.2.2, name the symbols file SPM$MANAGER_SYMBOLS_TUNE.COM.

If you are using the SPM$MANAGER_CONFIGURE.COM command procedure to create the SPM$MANAGER_SYMBOLS file, rename the symbols file to SPM$MANAGER_SYMBOLS_TUNE.COM after exiting the procedure.

4 Define the logical SPM$MANAGER_SYMBOLS_TUNE to point to the symbol file by placing the following command in the LOGIN.COM file of the account represented by the "value_of_spm_username" in the next command example.

```
$ DEFINE SPM$MANAGER_SYMBOLS_TUNE SPM$MANAGER_SYMBOLS_TUNE.COM
```

5 Submit SPM$MANAGER.COM to the batch queue by typing the following command:

```
$ SUBMIT /USER={value_of_spm_username}/NOPRINT/PARAMETER=("BOOT", -
         "TUNE")/QUE='NODE'{node_specific_batch_queue_extension} -
         {value_of_procedure_directory}SPM$MANAGER -
         /LOG={value_of_batchlog_directory}'NODE'_SPM$MANAGER.LOG
```

In the above command, TUNE is specified as the second parameter to specify a tuning collection process.

6 Select the method described in either Section 3.2.1 or Section 3.2.2 and create a symbols file for the SPM$COLLECT_CAPACITY process.

7 Specify SPM$COLLECT_CAPACITY as the value for the JOB_NAME symbol and a collection interval appropriate for historical reporting data for the COLLECTION_INTERVAL symbol. A value of 900 seconds is usually specified for historical reporting collections.

8 If you are creating a symbols file using the instructions in Section 3.2.2, name the symbols file SPM$MANAGER_SYMBOLS_CAPACITY.COM.

If you are using the SPM$MANAGER_CONFIGURE.COM command procedure to create the SPM$MANAGER_SYMBOLS file, rename the symbols file to SPM$MANAGER_SYMBOLS_CAPACITY.COM after exiting the procedure.

9 Define the logical SPM$MANAGER_SYMBOLS_CAPACITY to point to the file by placing the following command in the LOGIN.COM file of the account represented by the "value_of_spm_username" in the next command example.

```
$ DEFINE SPM$MANAGER_SYMBOLS_CAPACITY SPM$MANAGER_SYMBOLS_CAPACITY.COM
```

10 Submit SPM$MANAGER.COM to the batch queue by typing the following command:

```
$ SUBMIT /USER={value_of_spm_username}/NOPRINT -
         /PARAMETER=("BOOT","CAPACITY") -
         /QUE='NODE'{node_specific_batch_queue_extension} -
         {value_of_procedure_directory}SPM$MANAGER -
         /LOG={value_of_batchlog_directory}'NODE'_SPM$MANAGER.LOG
```

3.2.4 SPM$MANAGER.COM Error Reporting

SPM$MANAGER sends mail to the accounts specified in the DISTRIBUTION_LIST parameter of the SPM$MANAGER_SYMBOLS.COM whenever system threshold values are exceeded. SPM$MANAGER also sends mail when errors are encountered during the collection and reporting process.

When errors are reported, read the error description in the mail message. If you need further information, read the SPM$MANAGER batch log file in the batch log directory [.BATCHLOG] of the account from which SPM$MANAGER.COM was run.

3.2.5 Chapter Summary

You can collect performance data conventionally using SPM data collection commands, or automatically using the SPM$MANAGER.COM command procedure.

There are two commands for conventional data collection:

```
$ PERFORMANCE COLLECT=TUNE
$ PERFORMANCE COLLECT=CAPACITY
```

Both commands are capable of collecting the same types of data. These commands take one parameter to specify node name and a variety of qualifiers to tailor the collection process.

The SPM$MANAGER.COM command procedure automatically collects performance data. You can set up the environment in which the SPM$MANAGER.COM command procedure is to run in two ways. One way is using SPM$MANAGER_CONFIGURE.COM and the other is by following the instructions in Section 3.2.2 to create the SPM$MANAGER_SYMBOLS.COM

Once it is set up, the SPM$MANAGER.COM command procedure automatically collects data, evaluates the data, and notifies you if threshold values are exceeded on your system.

You may also set up the SPM$MANAGER.COM command procedure to run both SPM_TUNE and SPM_CAPACITY processes simultaneously.

Once you have started data collections, turn to Chapter 5 for an introduction to performance tuning, or Chapter 4 for an introduction to reporting.

4 Introduction to Reporting

This chapter introduces reporting by providing the following information:

- Description of the reporting commands
- Features of log file reporting
- Features of history file reporting

4.1 The Reporting Commands

VAX SPM provides two commands for reporting on performance data:

```
$ PERFORMANCE REPORT=LOG_FILE
$ PERFORMANCE REPORT=HISTORY
```

Both reporting commands take one parameter, the name of the log or history file, and a number of qualifiers. Use the REPORT=LOG_FILE command to report on data in a log file and the REPORT=HISTORY to report on data in a history file. A history file can contain data from many log files.

4.2 Features of Log File Reporting

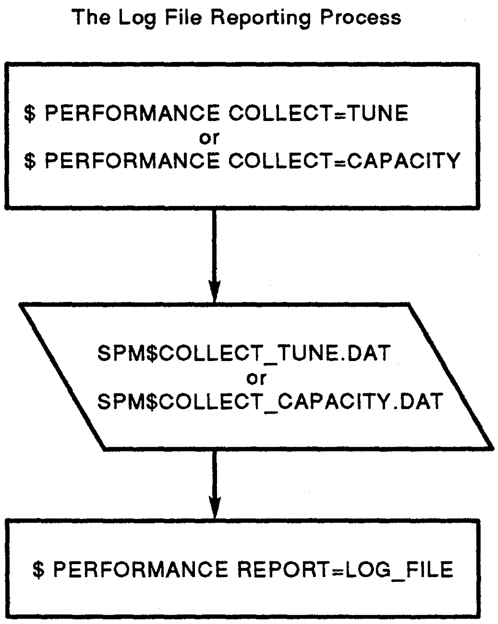
Figure 4–1 illustrates the log file reporting process.

As shown in Figure 4–1, data collected with either the COLLECT=TUNE or COLLECT=CAPACITY commands can be reported using the REPORT=LOG_FILE command. Although both commands are identical, the COLLECT=TUNE command is usually used to collect data for tuning purposes and the COLLECT=CAPACITY command is usually used to collect data for historical reporting purposes. Regardless of which command performs the collection, log files have the following characteristics:

- Usually contain data from one day
- Contain data from only one node
- Contain data collected at a short interval (for example, 5 minutes)

- Can contain (in addition to various performance metrics) the following configuration data:
  - System configuration and run statistics
  - Hardware configuration
  - Installed images
  - Global sections
  - System parameters

Figure 4–1 The Log File Reporting Process



```
$ PERFORMANCE COLLECT=TUNE
$ PERFORMANCE COLLECT=CAPACITY
SPM$COLLECT_TUNE.DAT
SPM$COLLECT_CAPACITY.DAT
$ PERFORMANCE REPORT=LOG_FILE
```

The REPORT=LOG_FILE command provides the following reporting and graphing options:

- Generate tabular reports and graphs of performance metrics.
- Generate working graphs of performance metrics in ANSI format.
- Generate presentation graphs of performance metrics in ReGIS or sixel format.
- Selectively inhibit the generation of graphs based upon performance thresholds you define.
- Dump the data used to generate reports and graphs into a file from which custom reports and graphs can be generated using a package such as DATATRIEVE or DECgraph.

Command qualifiers allow you to report or graph the following information:

- Various classes of performance metrics
- A reporting period within the log file
- Prime and/or nonprime hours of the day
- Statistics for the disks and devices you select

Reports and graphs generated with the REPORT=LOG_FILE command are most frequently used for performance tuning purposes.

This command produces a tabular report, a graph of CPU utilization for all records in a log file, and run statistics. The tabular report will contain final statistics for the default system metrics of CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE.

```
$ PERFORMANCE REPORT=LOGFILE SPM$COLLECT_TUNE.DAT /OUTPUT=REPORT.RPT
```

The following command produces the same tabular report and graph as in the above example; however, instead of covering the entire log file, it reports on records collected from 10 a.m. to 12 p.m.

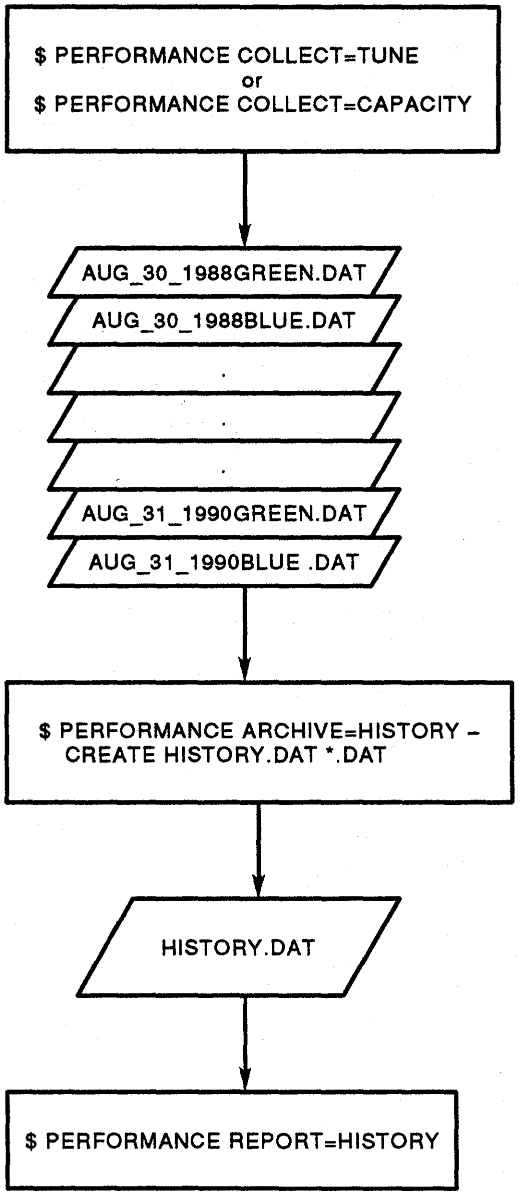
```
$ PERFORMANCE REPORT=LOG_FILE/SINCE=30-AUG-1988:10:00-
  /BEFORE=30-AUG-1988:12:00 SPM$COLLECT_TUNE.DAT /OUTPUT=REPORT.RPT
```

Chapter 14 describes how to generate reports and graphs from data in a log file.

4.3 Features of History File Reporting

Figure 4–2 illustrates the history file reporting process.

Figure 4–2 The History File Reporting Process



As shown in Figure 4–2, either the COLLECT=TUNE or COLLECT=CAPACITY command is first used to collect data into log files.

Data from a number of log files can be merged together into a history file by using the ARCHIVE command. A history file has the following characteristics:

- Can contain performance data for a number of years
- Can contain data for all nodes in a VAXcluster system
- Contains data archived at a relatively long interval (for example, 15 minutes)
- Contains various performance metrics but does not contain configuration data

The command below creates a history file from all log files with the extension .dat.

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE HISTORY.DAT *.DAT
```

Once you have created a history file, you can use the ARCHIVE command to maintain the history file. For example, you can add records from additional log and history files, delete records, and list the contents of the history file.

The following command lists information on records in the history file called HISTORY.DAT:

```
$ PERFORMANCE ARCHIVE=HISTORY/LIST HISTORY.DAT
```

The REPORT=HISTORY command provides the following graphing and reporting options:

- Generate tabular reports and graphs of performance metrics.
- Generate working graphs of performance metrics in ANSI format.
- Generate presentation graphs of performance metrics in ReGIS or sixel format.
- Selectively inhibit the generation of graphs based upon performance thresholds you define.
- Dump the data used to generate reports and graphs into a file from which custom reports and graphs can be generated by using a package such as DATATRIEVE or DECgraph.

Command qualifiers allow you to report or graph the following types of information:

- Various classes of performance metrics
- A reporting period within the history file
- Prime and/or nonprime hours of the day, days of the week, and dates you define
- Reporting units of days, weeks, and months
- User-defined reporting units, such as months in a fiscal quarter

- Data that is averaged
- Statistics for the disks and devices you select
- VAXcluster system data on a by node or a by cluster basis

Reports and graphs generated from history files by using the REPORT=HISTORY command are most frequently used for trend analysis and management reporting purposes. Reports from a history file can also be used in system tuning, however, to isolate the causes of disk contention within a VAXcluster system (see Chapter 8).

The following command produces a tabular report and a graph for a week beginning August 1 and ending August 7. A final statistics tabular report will be generated for each day of the week containing default system metrics of CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE. Each column on the graph represents CPU utilization statistics for one day of the week. The week is represented by the entire graph.

```
$ PERFORMANCE REPORT=HISTORY/SINCE=01-AUG-1988-
  /BEFORE=08-AUG-1988 HISTORY.DAT /OUTPUT=REPORT.RPT
```

The following command produces the same type of report and graph as the above command; however, the /SINCE and /BEFORE qualifiers specify a reporting period of one month and the /WEEKLY qualifier specifies a reporting unit of one week. A final statistics tabular report will be generated for each week in the month. The graph represents the entire month and each column represents a week in the month.

```
$ PERFORMANCE REPORT=HISTORY/WEEKLY/SINCE=01-AUG-1988-
  /BEFORE=01-SEP-1988 HISTORY.DAT /OUTPUT=REPORT.RPT
```

Chapter 15 describes how to create a history file from log files and Chapter 16 describes how to generate reports and graphs from data in a history file.

Chapter 17 describes presentation quality graphs and how to generate them.

Chapter 18 discusses using the REPORT=HISTORY command to perform trend analysis for a system.

Chapter 16 describes using the REPORT=HISTORY command for producing reports and graphs for a VAXcluster system.

5 Introduction to System Tuning

This chapter introduces system tuning using VAX SPM and provides the following information:

- Overview of the system tuning process
- Description of VAX SPM performance evaluation tools
- Prerequisites for system tuning

5.1 The Purpose of System Tuning

More than anyone else, a system manager has a vested interest in system performance. Whether to keep irate users from the door or to maintain confidence that the system is running at its optimal performance level, monitoring system performance is an important aspect of system management.

The most important benefit of a performance investigation is improved performance. A performance investigation reveals ways that a system manager can improve system performance: by adjusting the system workload or altering the value of system parameters. Users perceive these improvements, while VAX SPM reports and graphs verify and measure them.

A performance investigation provides additional benefits to the system manager, which are not as easily measured. For example, understanding system performance allows you to evaluate users' complaints with a clear perspective. You are better able to know when to reset their expectations and when to make modifications to the system. You know which system resources are saturated and under what workloads.

VAX SPM provides the following tools to help the system manager monitor, evaluate, and improve performance:

- A variety of reports for "hands-on" tuning
- The VAX SPM Performance Analyzer
- The Video Display utilities

These tools are used in the context of a tuning methodology based upon the information in the *Guide to VAX/VMS Performance Management*, Chapter 4.

5.2 Overview of System Tuning

The combined task of investigating system performance and making adjustments to improve system performance is called system tuning.

System tuning can be divided into the following four steps:

1. Defining the problem—Generate reports from collected data and isolate hot spots for further investigation.
2. Diagnosing and isolating the cause of the problem—Use the "hands-on" approach, the Analyzer, or video displays to determine the cause of a resource limitation.
3. Correcting the problem—This includes but is not limited to making changes to system parameters, rearranging workloads, and/or scheduling of system resources. Approaches to correcting performance problems are described in the *Guide to VAX/VMS Performance Management*.
4. Verifying the corrections—To verify the results of your performance tuning, reproduce the reports or displays that originally showed the problem. Use current data to verify that the problem no longer exists. You may use the hands-on approach, the Analyzer, or video displays to verify your corrections.

5.2.1 Goals of System Tuning

Goals

Before initiating system tuning, it is advisable to have a clear understanding of the goals and expectations of the tuning process. These goals and expectations are:

- System tuning addresses *chronic* (not *immediate*) problems. Do not tune a system for a one-time performance problem.
- System tuning helps to understand the relationship between the system workload and system capacity. No amount of tuning can compensate for a severe lack of resources.
- System tuning helps to understand performance complaints. Learn what is behind perceived performance degradation.
- System tuning shows when to tune a system to reallocate resources more effectively. Not all resource limitations can be remedied by system tuning.
- System tuning evaluates the effectiveness of a tuning operation and shows when to recognize success and when to stop.

Nongoals

The following are not reasonable goals or expectations of the tuning process:

- Capacity planning
- RMS tuning
- Application tuning
- DECnet VAX tuning

5.3 VAX SPM Tools for System Tuning

VAX SPM provides tools to help you evaluate your system's performance. A summary description of each tool and the chapter in which it is described is provided below.

5.3.1 Reports for Hands-On System Tuning

In the hands-on approach to system tuning, a variety of graphical and tabular reports are used. In this approach there are two parts to the tuning methodology: diagnosing the limiting resource and isolating the limiting resource. Diagnosing the limiting resource is accomplished by evaluating VAX SPM tabular reports and is described in Chapter 7. Isolating the limiting resource is accomplished by evaluating VAX SPM tabular reports, Process Metrics, and Extended Process Metrics reports as described in Chapter 8.

The VAX SPM Charge shows a detailed analysis of system resource use. The Charge utility helps you determine where to focus your tuning investigations by identifying the accounts that are using the most system resources. Chapter 20 describes the Charge utility.

If evidence of disk fragmentation is revealed while you are isolating resource limitations, use the Disk Space Analysis to determine on which disk fragmentation occurs. The Disk Space Analysis facility is described in Chapter 10.

If an I/O bottleneck can be attributed to high disk activity, use the File Activity display to identify the most active files on these disks. Chapter 9 describes the File Activity display.

If you need to take a closer look at system CPU activity while you are isolating resource limitations, use the System PC Analysis facility described in Chapter 11.

5.3.2 VAX SPM Analyzer

The VAX SPM Analyzer is an automated approach to the tuning methodology. It evaluates data in a VAX SPM log file to locate system bottlenecks and make recommendations on how to alleviate them. It has an interactive tutorial approach, which helps you to learn the tuning methodology while investigating system performance. You can use the Analyzer to confirm the results of your investigations using the tuning methodology or when the tuning methodology fails to identify any problems. The VAX SPM Analyzer evaluates performance quickly and produces reports that explain its results and recommendations. Chapter 13 describes the VAX SPM Analyzer.

5.3.3 Video Graphics Utility

The VAX SPM Video Graphics utility collects current performance data on a VMS system and displays it on a video terminal. Use the Video Graphics utility any time you wish to monitor or evaluate current system performance. There are two video graphics commands:

- PERFORMANCE DISPLAY=INVESTIGATE for the Investigate displays
- PERFORMANCE DISPLAY=RESOURCE for the Resource displays

The Investigate video graphics are designed to assist in tuning a single node system. The Resource video graphics are designed to evaluate performance for one or more nodes of a VAXcluster system.

Chapter 12 describes the VAX SPM Video Graphics utility.

5.4 Prerequisites to System Tuning

Complete the following tasks before tuning your system:

- Eliminate any known hardware problems. System tuning cannot compensate for hardware problems.
- Collect two weeks' worth of data to establish a baseline of current system performance. Use this baseline to evaluate improvements resulting from any adjustments you make to the system. Chapter 3 describes methods of data collection.

- Read the *Guide to VAX/VMS Performance Management*. This document provides an overview of system performance and makes specific recommendations for remedying and avoiding performance problems. The VAX SPM tuning methodology is based upon the principles of this book, which are adapted to make use of VAX SPM's comprehensive collection and reporting features.
- Interview users to learn their perceptions of system performance. Encourage them to be specific. For example, ask questions to determine how they were using the system, the exact times of performance degradation, what actually happened, and so on. You may find it helpful for users to telephone you when they notice problems on the system. You can then use the Video Graphics utility to monitor the system and note the times for further investigation at the end of the collection period.

To evaluate system performance using the hands-on approach or the Performance Analyzer, proceed to Chapter 6, "Starting System Tuning." To evaluate current system performance using the Video Graphic utility, proceed to Chapter 12, "Evaluating Performance Using Graphic Displays."

6 Starting System Tuning

This chapter describes the following steps required to begin tuning your system:

- Save and optimize system parameters.
- Begin collecting performance data.
- Generate and use the Hardware Configuration and System Parameters reports.
- Generate and interpret a System Summary graph.
- Isolate a "hot spot" for further investigation.

6.1 Saving and Optimizing System Parameters

The following procedure uses SYSGEN and AUTOGEN. SYSGEN saves the original values of system parameters so they can be restored if necessary during the tuning process. AUTOGEN sets the system parameters to their optimal values for your system's configuration.

It is strongly recommended that you perform this procedure before you start to tune your system. If AUTOGEN has been run recently on your system and you are confident that your system's parameters have appropriate values, you may want to perform only the SYSGEN portion of the procedure. If this is the case, omit steps 2 through 6 and perform only the first instruction to save a version of the original system parameters before proceeding to the next section.

1 Save the current parameter values in a parameter file and a list file by typing the following commands:

```
$ SET DEFAULT SYS$SYSTEM:
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE CURRENT
SYSGEN> WRITE nodename_BEFORE_yymmdd.PAR
SYSGEN> SET/OUTPUT=SYS$MANAGER:nodename_BEFORE_yymmdd.LIS
SYSGEN> SHOW/ALL
SYSGEN> SHOW/SPECIAL
SYSGEN> EXIT
$ COPY PARAMS.DAT,MODPARAMS.DAT SYS$MANAGER:*.BEFORE
```

2 Run AUTOGEN to establish optimal settings:

```
$ SET DEFAULT SYS$SYSTEM:
$ DELETE PARAMS.DAT.;*,MODPARAMS.DAT:*
$ @SYS$UPDATE:AUTOGEN SAVPARAMS SETPARAMS NOFEEDBACK
```

3 Save the values that AUTOGEN assigned to system parameters in a list file and reinstate the original values using the following commands:

```
$ SET DEFAULT SYS$SYSTEM:
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SET/OUTPUT=SYS$MANAGER:nodename_AFTER-yymmdd.LIS
SYSGEN> SHOW/ALL
SYSGEN> SHOW/SPECIAL
SYSGEN> USE nodename_BEFORE_yymmdd.PAR
SYSGEN> WRITE ACTIVE
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

4 Generate a difference file to identify differences between the original system parameter values and the ones that AUTOGEN assigned.

```
$ SET DEFAULT SYS$MANAGER:
$ DIFFERENCES/PARALLEL/OUTPUT=PARAMETERS.DIFF -
 nodename_BEFORE_yymmdd.LIS -
 nodename_AFTER_yymmdd.LIS
```

5 Review the differences between the system parameters and the ones that AUTOGEN assigned.

6 Make adjustments to system parameter values for GLBSECTIONS and GLBPAGES for layered products and applications. To make the best use of AUTOGEN's expertise, use the ADD_xxxx, MIN_xxxx, and MAX_xxxx forms for modifying parameter values rather than supplying absolute values.

Unless you have good reason, do not change the values of NPAGEDYN, PAGEDYN, LRPCOUNT, IRPCOUNT(V), SRPCOUNT(V), or any ACP_xxxx parameters.

6.2 Collecting Performance Data

It is recommended that you collect several days' worth of performance data to establish a performance baseline for your system. Use this performance baseline to measure the results of any adjustments you make to the system during the tuning process. Chapter 3 describes two methods for collecting data.

If you have not already begun collecting data and wish to begin tuning, select a time of high system activity, and collect an hour's worth of data using the following command:

```
$ PERFORMANCE COLLECT=TUNE
```

Stop the collection process using the following command:

```
$ PERFORMANCE COLLECT=TUNE/STOP
```

6.3 Generating the System Parameters and Hardware Configuration Reports

The System Parameters report is an alphabetized list of the system parameters and their values. The Hardware Configuration report lists all hardware for your VMS configuration. Both reports are useful during the tuning process.

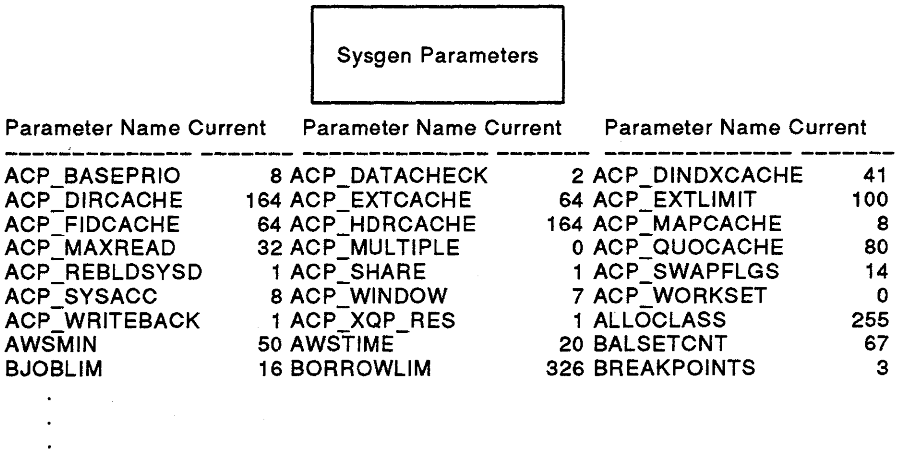
Use the command below to generate the System Parameters and Hardware Configuration reports. This command also produces a tabular report.

```
$PERFORMANCE REPORT=LOG_FILE/TABULAR -
/CLASS=(SYSGEN_PARAMETERS, HARDWARE_CONFIGURATION) SPM$COLLECT_TUNE.DAT
```

6.3.1 The System Parameters Report

You can use the System Parameters report in two ways: 1) generate this report before making any modifications to the system for a hard copy of the original system parameter values, and 2) generate it as necessary during your tuning investigation to use as a reference for current system parameter values. Figure 6–1 is an example of a System Parameters report segment.

Figure 6–1 System Parameters Report Segment



```
                         Sysgen Parameters

Parameter Name Current  Parameter Name Current   Parameter Name Current
-------------- ------- -------------- -------  -------------- -------
ACP_BASEPRIO         8 ACP_DATACHECK        2 ACP_DINDXCACHE      41
ACP_DIRCACHE       164 ACP_EXTCACHE        64 ACP_EXTLIMIT       100
ACP_FIDCACHE        64 ACP_HDRCACHE       164 ACP_MAPCACHE         8
ACP_MAXREAD         32 ACP_MULTIPLE         0 ACP_QUOCACHE        80
ACP_REBLDSYSD        1 ACP_SHARE            1 ACP_SWAPFLGS        14
ACP_SYSACC           8 ACP_WINDOW           7 ACP_WORKSET          0
ACP_WRITEBACK        1 ACP_XQP_RES          1 ALLOCLASS          255
AWSMIN              50 AWSTIME             20 BALSETCNT           67
BJOBLIM             16 BORROWLIM          326 BREAKPOINTS          3
   .
   .
   .
```

6.3.2 The Hardware Configuration Report

The Hardware Configuration report is useful when you are tuning a remote system or one with which you are unfamiliar and do not have access to the hardware information. You can also use the Hardware Configuration report to identify disks and devices for which you do not wish to collect data. Figure 6–2 is an example of a Hardware Configuration report segment.

Figure 6–2 Hardware Configuration Report Segment

```
                         +------------------------+
                         | Hardware Configuration |
                         +------------------------+

System System  Node    HW                   Control-
Id Hi   Id Lo  Name    Type Slot Adapter    ler        Unit  Class    Type        Size
-----  ------- -----   ---- ---- -------- -------- -----  -----    --------    -----
    0   20250  GREEN   VAX    0 UNKNOWN  CSA          1  DISK     RX50         800
                                                      2  DISK     RX50         800
                                         LTA          0 TERMINAL UNKNOWN      0
                                                   6969 TERMINAL LN03         0
                                         MBA          1  MAILBOX UNKNOWN      0
                                                      2  MAILBOX UNKNOWN      0
                                                      3  MAILBOX  LCLMEM MBX 0
                                                      4  MAILBOX  LCLMEM MBX 0
                                                      5  MAILBOX  LCLMEM MBX 0
                                                     56  MAILBOX  LCLMEM MBX 0
                                                     62  MAILBOX  LCLMEM MBX 0

                                  .
                                  .
                                  .
```

6.4 Generating a System Summary Graph

The System Summary graph provides a visual representation of system activity over time. Use the following command to generate System Summary graphs from the log files that contain collected data:

```
$ PERFORMANCE REPORT=LOG_FILE/GRAPH=SUMMARY/NOTABULAR SPM$COLLECT_TUNE.DAT
```

Figure 6–3 is an example segment of a System Summary graph. The vertical axis of the system summary graph measures activity as a percentage of system capacity. This means that the higher the column, the larger the percentage of system capacity used. The horizontal axis is calibrated to represent intervals within the collection period. Each column represents a collection interval, and every fifth column is labeled with the time of the activity represented in the column.

Figure 6–3 System Summary Graph Segment

```
                            System Summary (Percent) vs. Time of Day
                          Each column = 600 seconds /10.00 minutes
       From: 26-MAY-1988 07:56:25.73      To: 26-MAY-1988 17:05:00.17 \\
       ++----+----+----+----+----+----+----+----+----+----+----+-+--- //
100%   +                           I               IIIIIIIIIIII       \\
       |                          II               IIIIIIIIIIII       //
       |                          I*              IIIIIIIIIIIII       \\
       |                          **              IIIIIIIIIIIII       //
       |                          **              IIIIIIIIIIIII       \\
 90%   +                          **              IIIIIIIIIIIII       //
       |                          *C              IIIIIIIIIIIII       \\
       |                          *C              IIIIIIIIIIIII       //
       |                          *C              IIIIIIIIIIIII       \\
       |                          *C              IIIIIIIIIIIII       //
 80%   +                          *C              IIIIIIIIIIIII       \\
       |                     I    CC              IIIIIIIIIIIII       //
       |               *     I    CC              IIIIIIIIIIIII       \\
       |               C     *    CC              IIIIIIIIIIIII       //
       |               C     *    CC              IIIIIIIIIIIII       \\
 70%   +               C     *    CC              IIIIIIIIIIIII       //
       |               C  I  *    CC              IIIIIIIIIIIII       \\
       |               C  II *    CCII            IIIIIIIIIIIII       //
       |               C  II *    CCII            IIIIIIIIIIIII       \\
       |              IC  ** C    CCII            IIIIIIIIIIIII       //
 60%   +              IC  ** C    CC*I            IIIIIIIIIIIII       \\
       |              *C  ** C    CC*I         I  IIIIIIIIIIIII       //
       |              *C  *C C*  ICC*II        I  IIIIIIIIIIIII       \\
       |I             *C  *C C*  ICC*II        *  IIIIIIIIIIIII       //
       |I         I   CC  *C CC **CC*II        *  IIIIIIIIIIIII       \\
 50%   +I         I   CC  *C CC C*CC**I     I  *  IIIIIIIII*III       //
       |I         *   CCIICC CC C*CC***     I  *  I*IIIIIII*III       \\
       |I         *   CCIICC CC CCCCC**I    *  C  I*IIII*II*III       //
       |I         C   CC**CC CC CCCCC***    *  C  **I*II*II*III       \\
       |*         C   CC**CCICC CCCCC**C    *  C  **I*II*II*III       //
 40%   +*         C  ICC**CCICC CCCCC**C    C  C  **I*II*II*III       \\
       |*      I  C  ICCC*CC*CC CCCCC**C*   CI C  **I*II*II****       //
       |*      *  C  ICCCCCC*CCCCCCCC**CC   C* C I**I*II*I*****       \\
       |*      C  C  ICCCCCC*CCCCCCCC**CC   CCIC I****II*******       //
       |C      C IC  *CCCCCCCCCCCCCCC**CC   CC*CII****II*******       \\
 30%   +C      C *C  *CCCCCCCCCCCCCCC**CC*  CCCC*******I*******       //
       |C      C CC  CCCCCCCCCCCCCCCC**CCC  CCCCC******I*******       \\
       |CC     CCCCC CCCCCCCCCCCCCCCC*CCCC  CCCCC******I*******       //
       |CC C I CCCCCICCCCCCCCCCCCCCC*CCCCC CCCCC**************       \\
       |CC C CICCCCC*CCCCCCCCCCCCCCCC*CCCCC CCCCCC*************       //
 20%   +CCCCCC*CCCCC*CCCCCCCCCCCCCCCC*CCCCCICCCCCC*************       \\
                            .      .      .
 10%   +CCCCCCCCCCCCCCCCCCCCCCCCCCCCC*CCCCCCCCCCCCC************       //
       |CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC************       \\
       |CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC************       //
       |CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC************       \\
       |CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC************.......//
       +-----+----+----+----+----+----+----+----+----+----+----+-+----\\
       07:56     09:36     11:16     12:57     14:37     16:17
          08:46     10:26     12:07     13:47     15:27      16:27
"C" = CPU Only   "*" = CPU & I/O   "I" = I/O Only  "." = Data Unavailable
```

6.5 Isolating a "Hot Spot"

A "hot spot" is a period of high system activity. Hot spots show where to focus your tuning investigations to locate resource limitations.

To isolate a hot spot, examine the System Summary graph and look for columns that approach the top of the graph, close to or beyond the 80% mark. Look at the bottom of the graph and note the times associated with these columns.

For example, in Figure 6–3, columns that reach the top of the graph or 100% of system capacity are for times between 14:37 and 16:27. Therefore, the hot spot for the graph is between 14:37 and 16:27.

Specify hot spots with the /SINCE and /BEFORE qualifiers, and use the following format:

```
DD-MMM-YYYY:HH:MM
```

The hot spot in Figure 6–3 would be expressed as follows:

```
/SINCE="26-MAY-1988:14:37"/BEFORE="26-MAY-1988:16:27"
```

If you are using the Analyzer to identify the resource limitation, you would use the following command to identify the hot spot in this example for analysis:

```
$ PERFORMANCE ANALYZE=LOG_FILE/SINCE="26-MAY-1988:14:37" -
/BEFORE="26-MAY-1988:16:27" SPM$COLLECT_TUNE.DAT
```

If you are using the hands-on approach to identify the resource limitation, you would use the following command to generate a tabular report for the hot spot in this example:

```
$ PERFORMANCE REPORT=LOG_FILE/TABULAR-
/SINCE="26-MAY-1988:14:37"/BEFORE="26-MAY-1988:16:27" SPM$COLLECT_TUNE.DAT
```

When you have identified hot spots, you can proceed to diagnose the limiting resource. If you are using the Analyzer, turn to Chapter 13. If you are using the hands-on approach, turn to Chapter 7.

7 Diagnosing a Resource Limitation

This chapter describes the hands-on approach to system tuning, and provides the following information about the tuning methodology:

- Overview of the tuning methodology
- Instructions for generating and using interval tabular reports
- Instructions for diagnosing a resource on your system that is limiting

7.1 Overview

The tuning methodology is an orderly approach used by the system manager or performance analyst to evaluate VAX SPM reports. This evaluation aims to identify limitations in the three major resources: memory, I/O, and CPU.

You must first identify a hot spot for your system as described in Chapter 6. The tuning methodology then consists of four steps:

1 Diagnose the limiting resource.

2 Isolate the limiting resource.

3 Correct the problem.

4 Verify the results.

To diagnose whether memory, I/O, or CPU is the limiting resource, apply manual tests to the values of fields in the tabular report. These tests are described in this chapter.

Once you have diagnosed the limiting resource or resources, apply further tests to isolate the cause of the limitation. Directions for isolating the cause of limitations are in Chapter 8, Chapter 10, and Chapter 11.

Resources are evaluated in the same order in both the diagnose and isolate steps of the tuning methodology: memory, I/O, and CPU. This order is important because a memory limitation may increase the demand on the other resources, causing the symptoms of an I/O or CPU limitation. When you diagnose a resource that is limiting, further investigate that resource.

Once you have isolated the cause of the resource limitation, apply the approaches described in the *VAX/VMS Guide to Performance Management* to correct the problem.

After correcting the problem, verify the results by collecting new data and generating the reports that initially displayed the problem to ensure it no longer exists.

If you encounter another limitation during verification, identify a hot spot and proceed with the tuning methodology until no resources are diagnosed as limiting.

Table 7–1 lists the VAX SPM tools to use in each step of the tuning methodology.

Table 7–1 VAX SPM Tools Used in the Tuning Methodology

Step in Tuning Methodology	Related Tool
Identify a "hot spot" (see Chapter 6)	System Summary Graph
Diagnose the Limiting Resource (Memory, I/O, CPU)	Tabular Report
	Process Metrics Report
	System Parameters Report
	Scheduler States Report
Isolate the Limiting Resource (Memory, I/O, CPU)	Tabular Report
	Process Metrics Report
	Extended Process Metrics
	Disk Space Report
	System PC Analysis Report
Apply the Remedy	*VAX/VMS Guide to Performance Management*
Verify the Results	System Summary Graph
	Tabular Report
	Any report that originally showed the problem can be run again with new data to verify that the problem has been corrected

If you are unable to identify a resource limitation after analyzing your performance data, one of the following situations may be true:

- There may be no problem.
- Users may have improper expectations of available resources.
- You may have made an error in your analysis.

If you do identify a limitation but are unable to correct it, one of these situations may be true:

- There may be insufficient resource capacity for the desired response time and throughput.
- Applications may be using system resources in an inefficient manner.

Whatever the outcome of your anlaysis, you may benefit from using the VAX SPM Analyzer and comparing the results with your evaluations. See Chapter 13 to learn how to use the VAX SPM Analyzer.

For some systems, tuning cannot help and the only effective way to remove the limitation is to acquire more of the resource. This means acquiring more memory, more peripherals, and perhaps a more powerful or additional CPU.

7.2 Generating Interval Tabular Reports

The tabular report is most often used in the tuning methodology. VAX SPM provides two types of tabular reports: interval and final. They are specified using the /INTERVAL and /TABULAR qualifiers for the PERFORMANCE REPORT=LOG_FILE command.

A Final Tabular report is a single report that averages interval statistics for a reporting period. An Interval Tabular report contains statistics for each interval in the reporting period.

Final Tabular reports are most commonly used for system tuning. There are two circumstances, however, where Interval Tabular reports are useful: when isolating the cause of excessive image activations (described in Chapter 8) and when investigating a hot spot that includes varying levels of activity. Use the Interval Tabular reports for a detailed view of system activity for these intervals.

The following command generates Interval Tabular reports:

```
$ PERFORMANCE REPORT=LOG_FILE/TABULAR=INTERVAL -
/BEFORE=beginning-time of extreme activity within hot spot -
/SINCE=end-time of extreme activity within hot spot -
SPM$COLLECT_TUNE.DAT
```

Note: Interval Tabular reports contain statistics for each interval within the reporting period, and may produce a significant amount of output.

7.3 Examining Resources

After identifying performance hot spots, you are ready to apply the tuning methodology to diagnose the resources that may be limiting performance on your system.

Use the following command to produce reports for an identified hot spot.

```
$ PERFORMANCE REPORT=LOG_FILE/TABULAR=FINAL -
/CLASS=(SCHEDULER,SYSGEN_PARAMETERS) -
/BEFORE=beginning-time-of-hot-spot -
/SINCE=end-time-of-hot-spot  SPM$COLLECT_TUNE.DAT
```

The above command produces three reports: the Final Tabular, System Parameters, and Scheduler States reports. Most of the tests in the following sections refer to fields in the tabular report; however, there are two tests that need information from the System Parameters and Scheduler States reports.

As explained earlier, diagnose the limiting resource by applying tests or rules for the three main resources: memory, I/O, and CPU (in that order). The rules for each resource are found in the tables listed below:

Resource	**Diagnosis Table**
Memory	Table 7–3
I/O	Table 7–4
CPU	Table 7–5

Some of the rules refer to a CPU factor. To apply these rules, refer to Table 7–2, CPU Factor by Processor. Locate your CPU type and note the CPU factor associated with it. Multiply this factor by the number given in the rule. Compare the result to the value in the rule.

When you find a rule that is true, note the resource and proceed to Chapter 8 to further investigate the cause of the limiting resource.

Table 7–2 CPU Factor by Processor

VAX Processor	CPU Factor	VAX Processor	CPU Factor
6210	3.0	8810	4.9
6220	5.2	8820	8.0
6230	7.4	8830	10.0
6240	8.2	8840	11.0
8200	0.8	11/725	0.3
8250	1.1	11/730	0.3
8300	1.4	11/750	0.65
8350	1.7	11/780	1.0
8530	3.5	11/785	1.4
8550	4.9	MicroVAX II	0.75
8600	3.4	MicroVAX 2000	0.60
8650	4.9	MicroVAX 3500	3.0
8700	4.9	MicroVAX 3600	3.0
8800	8.0		

Note: Numerical values provided as part of the rules in the following sections are guidelines. You may need to modify them for your workload. Likewise, CPU factors in Table 7–2 are intended as guidelines for diagnosing resource limitations on your processor and do not imply relative CPU speeds or performance. For tuning purposes these numbers are intentionally conservative, especially in the case of multiple CPUs.

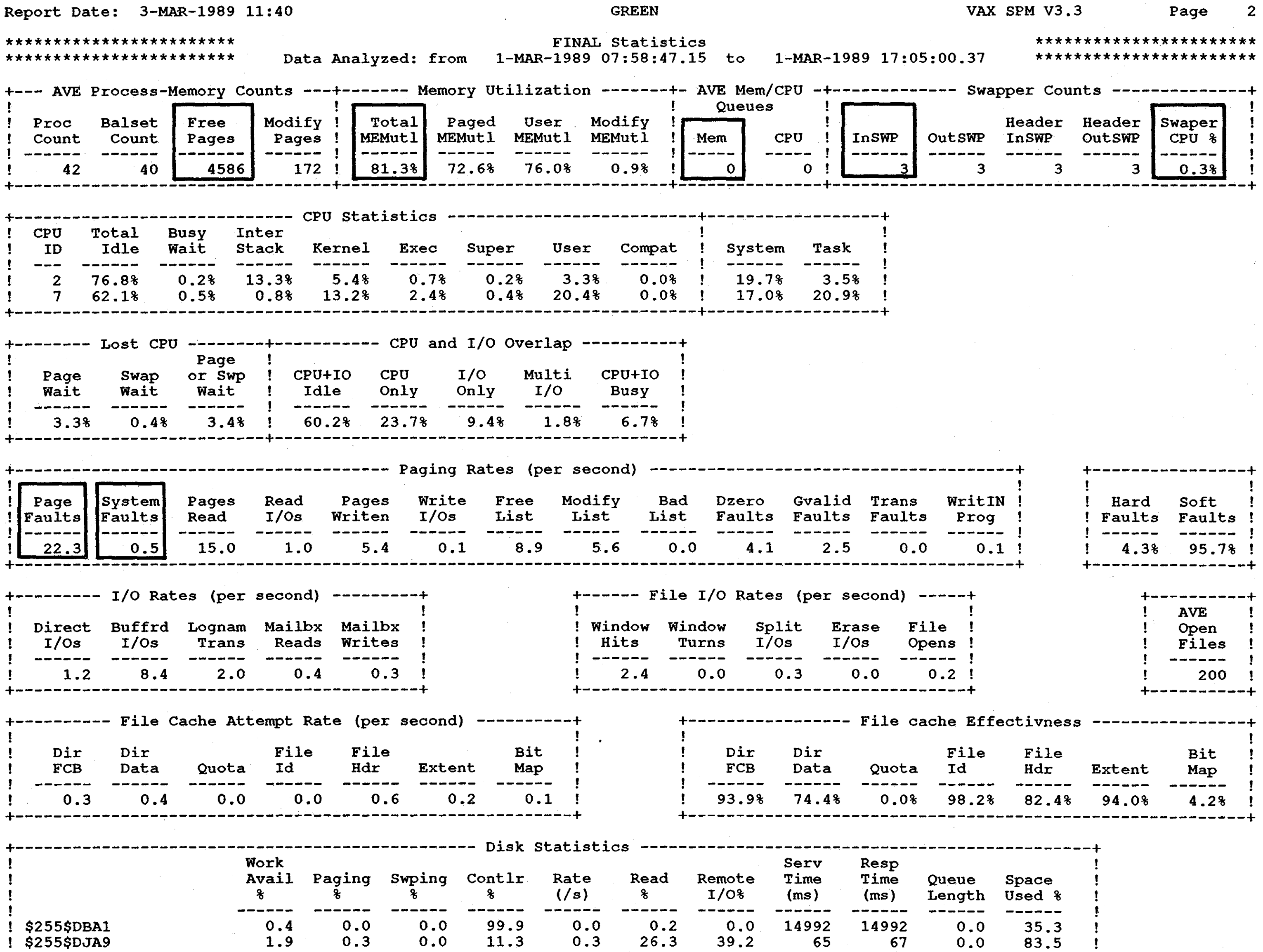
7.3.1 Diagnosing a Memory Limitation

Figure 7–1 is a sample tabular report similar to the one generated by the command in Section 7.3. In this report, the metrics related to diagnosing a memory limitation are highlighted. To diagnose a memory limitation, examine your tabular report using Figure 7–1 as a reference and apply the rules in Table 7–3.

For example, the first rule in Table 7–3 is:

```
Free Pages < GROWLIM (System Parameter)
```

To apply this rule, find the value for Free Pages on your tabular report and compare it to the value for the system parameter GROWLIM in your System Parameters report. If the value for Free Pages is less than the value for GROWLIM, there may be a memory limitation caused by lack of free memory and you would proceed to Chapter 8 to further investigate this limitation. Otherwise, this rule does not indicate a memory limitation and you would apply the next rule. If no rules are true, proceed to Section 7.3.2.



```
Report Date:  3-MAR-1989 11:40                                 GREEN                                VAX SPM V3.3         Page    2

************************                                 FINAL Statistics                                  ***********************
************************     Data Analyzed: from   1-MAR-1989 07:58:47.15  to   1-MAR-1989 17:05:00.37     ***********************

+--- AVE Process-Memory Counts ---+------- Memory Utilization -------+- AVE Mem/CPU -+------------- Swapper Counts -------------+
!                                 !                                  !    Queues     !                                           !
!  Proc   Balset   Free    Modify !   Total   Paged   User   Modify  !               !                  Header  Header  Swaper   !
!  Count   Count   Pages    Pages !  MEMutl  MEMutl  MEMutl  MEMutl  !  Mem     CPU  !  InSWP   OutSWP  InSWP   OutSWP   CPU %   !
! ------  ------  ------   ------ !  ------  ------  ------  ------  ! ------  ----- !  ------  ------  ------  ------  ------   !
!     42      40    4586      172 !   81.3%   72.6%   76.0%    0.9%  !      0      0 !       3       3       3       3    0.3%   !
+---------------------------------+----------------------------------+---------------+-------------------------------------------+

+----------------------------- CPU Statistics --------------------------+-------------------+
!  CPU   Total   Busy   Inter                                           !                   !
!   ID    Idle   Wait   Stack   Kernel   Exec   Super    User   Compat  !  System   Task    !
!  ---  ------  ------  ------  ------  ------  ------  ------  ------  !  ------  ------   !
!    2   76.8%    0.2%   13.3%    5.4%    0.7%    0.2%    3.3%    0.0%  !   19.7%    3.5%   !
!    7   62.1%    0.5%    0.8%   13.2%    2.4%    0.4%   20.4%    0.0%  !   17.0%   20.9%   !
+-----------------------------------------------------------------------+-------------------+

+-------- Lost CPU --------+----------- CPU and I/O Overlap -----------+
!                   Page   !                                           !
!   Page    Swap   or Swp  !  CPU+IO   CPU     I/O    Multi   CPU+IO   !
!   Wait    Wait    Wait   !   Idle    Only    Only    I/O     Busy    !
!  ------  ------  ------  !  ------  ------  ------  ------  ------   !
!    3.3%    0.4%    3.4%  !  60.2%    23.7%   9.4%    1.8%    6.7%    !
+--------------------------+-------------------------------------------+

+--------------------------------------- Paging Rates (per second) --------------------------------------+      +----------------+
!                                                                                                        !      !                !
!  Page   System   Pages   Read    Pages   Write   Free   Modify    Bad   Dzero   Gvalid  Trans   WritIN !      !  Hard   Soft   !
! Faults  Faults   Read    I/Os   Writen   I/Os    List    List    List   Faults  Faults  Faults   Prog  !      ! Faults  Faults !
! ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------ !      ! ------  ------ !
!   22.3     0.5    15.0     1.0     5.4     0.1     8.9     5.6     0.0     4.1     2.5     0.0     0.1 !      !   4.3%   95.7% !
+--------------------------------------------------------------------------------------------------------+      +----------------+

+--------- I/O Rates (per second) ---------+              +------ File I/O Rates (per second) ------+                +----------+
!                                          !              !                                         !                !  AVE     !
!  Direct  Buffrd  Lognam  Mailbx  Mailbx  !              ! Window  Window   Split   Erase   File   !                !  Open    !
!   I/Os    I/Os    Trans   Reads  Writes  !              !  Hits    Turns   I/Os    I/Os    Opens  !                !  Files   !
!  ------  ------  ------  ------  ------  !              ! ------  ------  ------  ------  ------  !                ! ------   !
!     1.2     8.4     2.0     0.4     0.3  !              !    2.4     0.0     0.3     0.0     0.2  !                !    200   !
+------------------------------------------+              +-----------------------------------------+                +----------+

+---------- File Cache Attempt Rate (per second) ----------+         +---------------- File cache Effectivness -----------------+
!                                                          !         !                                                         !
!    Dir    Dir             File    File             Bit   !         !    Dir    Dir             File    File             Bit   !
!    FCB    Data    Quota   Id      Hdr    Extent    Map   !         !    FCB    Data    Quota   Id      Hdr    Extent    Map   !
!  ------  ------  ------  ------  ------  ------  ------  !         !  ------  ------  ------  ------  ------  ------  ------  !
!     0.3     0.4     0.0     0.0     0.6     0.2     0.1  !         !   93.9%   74.4%    0.0%   98.2%   82.4%   94.0%    4.2%  !
+----------------------------------------------------------+         +---------------------------------------------------------+

+------------------------------------------------ Disk Statistics -----------------------------------------------+
!                        Work                                                    Serv    Resp                    !
!                        Avail  Paging  Swping  Contlr   Rate    Read   Remote   Time    Time   Queue   Space    !
!                         %       %       %       %      (/s)     %     I/O%    (ms)    (ms)   Length  Used %   !
!                       ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------   !
! $255$DBA1                0.4     0.0     0.0    99.9     0.0     0.2     0.0   14992   14992     0.0    35.3   !
! $255$DJA9                1.9     0.3     0.0    11.3     0.3    26.3    39.2      65      67     0.0    83.5   !
```

Figure 7–1 Tabular Report for Diagnosing a Memory Limitation

Table 7–3 Diagnosing a Memory Limitation

Rule	Probable Cause
Free Pages < GROWLIM (System Parameter)	No Free Memory
Total MEMutl ≥ 90%	No Free Memory
Page Faults ≥ 100 × (CPU Factor)[1]	High Page Faults
System Faults > 1	High Page Faults
AVE Mem Queue > 0	High Swapping
InSWP > 0	High Swapping
Swaper CPU% ≥ 2	High Swapping

[1] See Table 7–2 for the CPU factor associated with your CPU.

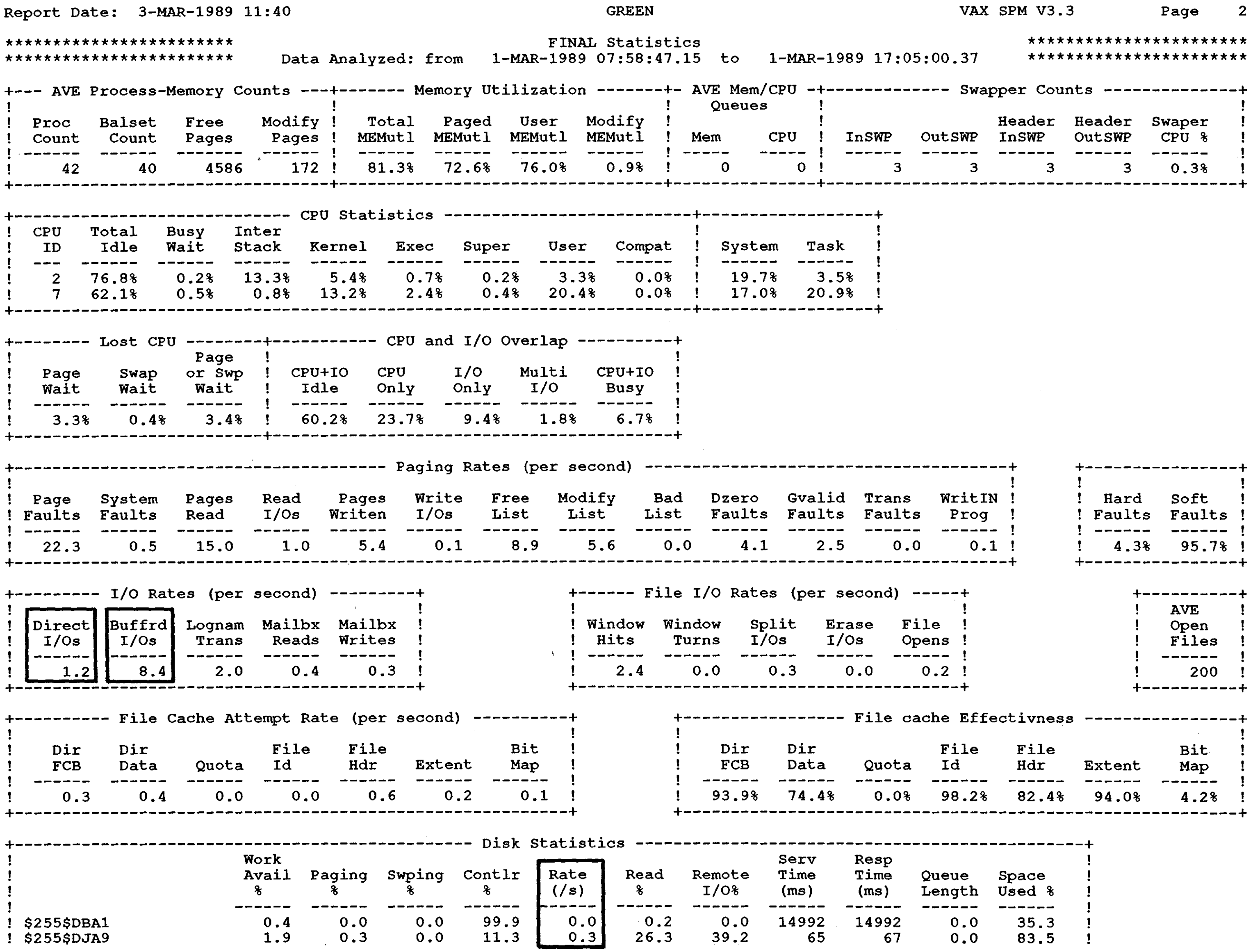
7.3.2 Diagnosing an I/O Limitation

Figure 7–2 is a sample tabular report in which the fields related to diagnosing an I/O limitation are highlighted. Using Figure 7–2 as a reference, apply the rules in Table 7–4 to fields in your tabular report in order to diagnose an I/O limitation.

For example, the first rule in Table 7–4 is:

```
Direct I/Os ≥  30 ×  (CPU Factor)†
```

The dagger (†) at the end of the rule refers you to Table 7–2 to determine the CPU factor. For example, if your processor type is 8600, the CPU factor is 3.4. Multiply this factor by 30 to obtain 102. Now compare the value for Direct I/Os on your tabular report with 102. If the value of Direct I/Os is equal to or greater than 102, you may have an I/O limitation and you would proceed to Chapter 8. If the value of Direct I/Os is less than 102, proceed to the next test. If none of the rules are true, proceed to Section 7.3.3.



```
Report Date:  3-MAR-1989 11:40                    GREEN                             VAX SPM V3.3         Page     2

************************                         FINAL Statistics                             ***********************
************************    Data Analyzed: from   1-MAR-1989 07:58:47.15  to   1-MAR-1989 17:05:00.37     ***********************

+--- AVE Process-Memory Counts ---+------- Memory Utilization -------+- AVE Mem/CPU -+------------ Swapper Counts -------------+
!                                 !                                  !     Queues    !                                         !
!  Proc    Balset   Free   Modify !  Total   Paged   User    Modify  !               !                   Header  Header  Swaper  !
!  Count   Count    Pages   Pages !  MEMutl  MEMutl  MEMutl  MEMutl  !  Mem      CPU !  InSWP   OutSWP   InSWP   OutSWP  CPU %   !
!  ------  ------  ------- ------ !  ------  ------  ------  ------  !  -----  ----- !  ------  ------   ------  ------  ------  !
!    42      40     4586     172  !  81.3%   72.6%   76.0%    0.9%   !    0        0 !     3       3        3       3     0.3%  !
+---------------------------------+----------------------------------+---------------+-----------------------------------------+

+----------------------------- CPU Statistics -------------------------+-----------------+
!  CPU   Total   Busy    Inter                                           !                 !
!   ID   Idle    Wait    Stack   Kernel   Exec    Super   User   Compat !  System   Task  !
!  ---  ------  ------  ------   ------  ------  ------  ------  ------ !  ------  ------ !
!   2   76.8%    0.2%   13.3%     5.4%    0.7%    0.2%    3.3%    0.0%  !   19.7%    3.5% !
!   7   62.1%    0.5%    0.8%    13.2%    2.4%    0.4%   20.4%    0.0%  !   17.0%   20.9% !
+----------------------------------------------------------------------+-----------------+

+-------- Lost CPU --------+----------- CPU and I/O Overlap ----------+
!                   Page   !                                          !
!  Page    Swap    or Swp  !  CPU+IO   CPU     I/O     Multi   CPU+IO !
!  Wait    Wait     Wait   !  Idle     Only    Only    I/O     Busy   !
!  ------  ------  ------  !  ------  ------  ------  ------  ------  !
!   3.3%    0.4%    3.4%   !  60.2%   23.7%    9.4%    1.8%    6.7%   !
+--------------------------+------------------------------------------+

+------------------------------------- Paging Rates (per second) ------------------------------------+      +---------------+
!                                                                                                      !      !               !
!  Page  System   Pages   Read    Pages   Write   Free   Modify   Bad    Dzero  Gvalid  Trans  WritIN !      !  Hard   Soft  !
! Faults Faults   Read    I/Os   Writen   I/Os    List    List    List   Faults Faults  Faults  Prog  !      ! Faults Faults !
! ------ ------  ------  ------  ------  ------  ------  ------  ------  ------ ------  ------ ------  !      ! ------ ------ !
!  22.3    0.5    15.0     1.0     5.4     0.1     8.9     5.6     0.0     4.1    2.5     0.0    0.1  !      !  4.3%  95.7%  !
+------------------------------------------------------------------------------------------------------+      +---------------+

+--------- I/O Rates (per second) ---------+                 +------ File I/O Rates (per second) -----+        +----------+
!                                          !                 !                                        !        !   AVE    !
!  Direct  Buffrd  Lognam  Mailbx  Mailbx  !                 !  Window  Window  Split   Erase   File  !        !   Open   !
!   I/Os    I/Os   Trans   Reads   Writes  !                 !   Hits   Turns   I/Os    I/Os   Opens  !        !   Files  !
!  ------  ------  ------  ------  ------  !                 !  ------  ------  ------  ------ ------ !        !  ------  !
!     1.2     8.4     2.0     0.4     0.3  !                 !    2.4     0.0     0.3     0.0    0.2  !        !    200   !
+------------------------------------------+                 +----------------------------------------+        +----------+

+---------- File Cache Attempt Rate (per second) ----------+       +---------------- File cache Effectivness ----------------+
!                                                          !       !                                                         !
!   Dir     Dir             File    File            Bit    !       !   Dir     Dir             File    File            Bit   !
!   FCB     Data    Quota   Id      Hdr     Extent  Map    !       !   FCB     Data    Quota   Id      Hdr     Extent  Map   !
!  ------  ------  ------  ------  ------  ------  ------  !       !  ------  ------  ------  ------  ------  ------  ------ !
!    0.3     0.4     0.0     0.0     0.6     0.2     0.1   !       !  93.9%   74.4%    0.0%   98.2%   82.4%   94.0%    4.2%  !
+----------------------------------------------------------+       +---------------------------------------------------------+

+--------------------------------------------- Disk Statistics --------------------------------------------+
!                          Work                                                    Serv    Resp                    !
!                          Avail   Paging  Swping  Contlr   Rate    Read   Remote   Time    Time   Queue   Space   !
!                           %        %       %       %      (/s)     %     I/O%     (ms)    (ms)   Length  Used %  !
!                         ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------   !
! $255$DBA1                  0.4     0.0     0.0    99.9     0.0     0.2     0.0   14992   14992     0.0    35.3   !
! $255$DJA9                  1.9     0.3     0.0    11.3     0.3    26.3    39.2      65      67     0.0    83.5   !
```

Figure 7–2 Tabular Report for Diagnosing an I/O Limitation

Table 7–4 Diagnosing an I/O Limitation

Rule	Probable Cause
Direct I/Os $\geq$ 30 × (CPU Factor)[1]	High Direct I/O
Rate(/s) > 15 for any disk	High Direct I/O
Buffrd I/Os $\geq$ 100 ×(CPU Factor)[1]	High Buffered I/O

[1] See Table 7–2 for the CPU factor associated with your CPU.

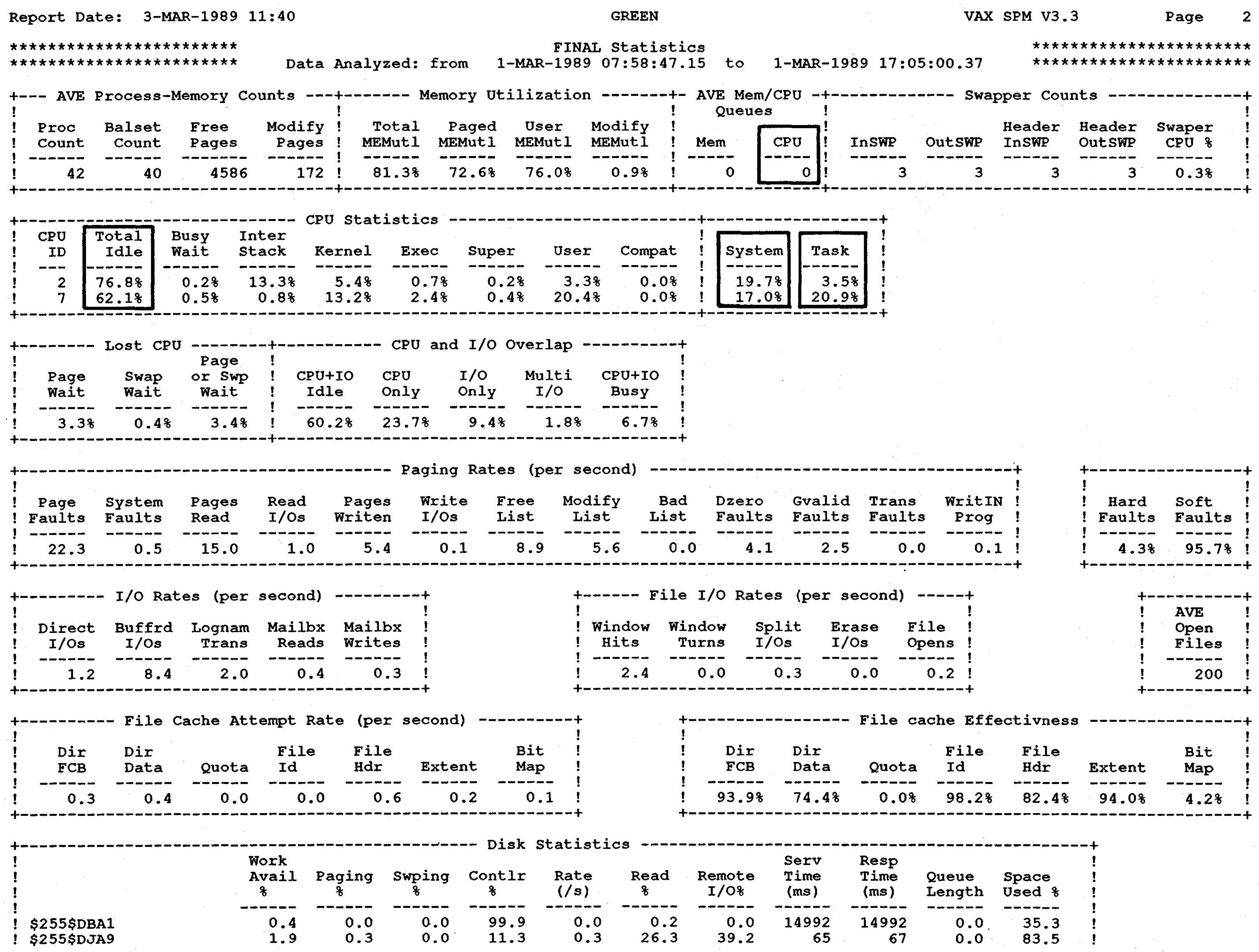
7.3.3 Diagnosing a CPU Limitation

Figure 7–3 is a sample tabular report in which the fields related to diagnosing a CPU limitation are highlighted. Figure 7–4 is an example of the scheduler states section of the tabular report. Using Figure 7–3 and Figure 7–4 as references, apply the rules in Table 7–5 to the corresponding fields in your tabular report to diagnose a CPU limitation.

For example, the fourth rule in Table 7–5 is:

```
System > Task and Task > 25% of the total CPU time
```

Look at the values for System and Task CPU time in your tabular report. If System CPU time is greater than Task CPU time, and Task is greater than 25% of the total CPU time, these values indicate a high system CPU time and a probable CPU limitation. Proceed to Chapter 8 to further investigate the CPU limitation. If the rule is not true, proceed to the next test.



```
Report Date:  3-MAR-1989 11:40                          GREEN                           VAX SPM V3.3        Page    2

************************                         FINAL Statistics                          ***********************
************************  Data Analyzed: from   1-MAR-1989 07:58:47.15  to   1-MAR-1989 17:05:00.37   ***********************

+--- AVE Process-Memory Counts ---+------- Memory Utilization -------+- AVE Mem/CPU -+------------ Swapper Counts -------------+
!                                 !                                  !    Queues     !                                         !
!  Proc   Balset   Free   Modify  !  Total   Paged    User   Modify  !               !                  Header  Header  Swaper   !
!  Count   Count   Pages   Pages  !  MEMutl  MEMutl  MEMutl  MEMutl  !  Mem     CPU  !  InSWP  OutSWP   InSWP   OutSWP  CPU %    !
!  ------  ------  ------  ------ !  ------  ------  ------  ------  !  -----  ----- !  ------  ------  ------  ------  ------   !
!     42      40    4586     172  !  81.3%   72.6%   76.0%    0.9%   !     0       0 !       3       3       3       3   0.3%    !
+---------------------------------+----------------------------------+---------------+-----------------------------------------+

+---------------------------- CPU Statistics ---------------------------+-------------------+
!  CPU   Total   Busy   Inter                                          !                   !
!  ID    Idle    Wait   Stack   Kernel   Exec    Super    User  Compat  !  System   Task    !
!  ---  ------  ------  ------  ------  ------  ------  ------  ------  !  ------  ------   !
!   2   76.8%    0.2%   13.3%    5.4%    0.7%    0.2%    3.3%    0.0%   !   19.7%    3.5%   !
!   7   62.1%    0.5%    0.8%   13.2%    2.4%    0.4%   20.4%    0.0%   !   17.0%   20.9%   !
+-----------------------------------------------------------------------+-------------------+

+-------- Lost CPU --------+----------- CPU and I/O Overlap ----------+
!                   Page   !                                         !
!  Page    Swap   or Swp   !  CPU+IO    CPU     I/O    Multi  CPU+IO  !
!  Wait    Wait    Wait    !  Idle     Only    Only     I/O    Busy   !
!  ------  ------  ------  !  ------  ------  ------  ------  ------  !
!   3.3%    0.4%    3.4%   !  60.2%   23.7%    9.4%    1.8%    6.7%   !
+--------------------------+-----------------------------------------+

+------------------------------------- Paging Rates (per second) -------------------------------------+   +---------------+
!                                                                                                     !   !               !
!  Page  System  Pages   Read   Pages   Write   Free  Modify    Bad  Dzero  Gvalid   Trans  WritIN     !   !  Hard   Soft   !
! Faults Faults  Read    I/Os   Writen  I/Os    List   List    List Faults  Faults  Faults   Prog     !   ! Faults Faults  !
! ------ ------ ------  ------  ------ ------  ------ ------  ------ ------  ------  ------  ------    !   ! ------ ------  !
!  22.3    0.5   15.0     1.0     5.4    0.1     8.9    5.6     0.0    4.1     2.5     0.0     0.1    !   !  4.3%   95.7%  !
+-----------------------------------------------------------------------------------------------------+   +---------------+

+--------- I/O Rates (per second) ---------+          +------ File I/O Rates (per second) -----+        +----------+
!                                          !          !                                        !        !   AVE    !
!  Direct  Buffrd  Lognam  Mailbx  Mailbx  !          ! Window  Window   Split   Erase   File  !        !   Open   !
!   I/Os    I/Os   Trans   Reads   Writes  !          !  Hits    Turns   I/Os    I/Os   Opens  !        !   Files  !
!  ------  ------  ------  ------  ------  !          ! ------  ------  ------  ------  ------ !        !  ------  !
!    1.2     8.4     2.0     0.4     0.3   !          !   2.4     0.0     0.3     0.0     0.2  !        !    200   !
+------------------------------------------+          +----------------------------------------+        +----------+

+---------- File Cache Attempt Rate (per second) ----------+      +---------------- File cache Effectivness ----------------+
!                                                          !      !                                                        !
!   Dir     Dir             File    File             Bit   !      !   Dir     Dir             File    File            Bit   !
!   FCB     Data   Quota     Id      Hdr    Extent   Map   !      !   FCB     Data   Quota     Id      Hdr   Extent   Map   !
!  ------  ------  ------  ------  ------  ------  ------  !      !  ------  ------  ------  ------  ------  ------  ------ !
!    0.3     0.4     0.0     0.0     0.6     0.2     0.1   !      !  93.9%   74.4%    0.0%   98.2%   82.4%   94.0%    4.2% !
+----------------------------------------------------------+      +--------------------------------------------------------+

+------------------------------------------- Disk Statistics -------------------------------------------+
!                          Work                                                Serv    Resp               !
!                          Avail   Paging  Swping  Contlr   Rate   Read   Remote  Time    Time   Queue  Space  !
!                            %       %       %       %      (/s)     %     I/O%   (ms)    (ms)  Length Used %  !
!                          ------  ------  ------  ------  ------ ------  ------ ------  ------ ------ ------  !
! $255$DBA1                  0.4     0.0     0.0    99.9     0.0    0.2     0.0  14992   14992    0.0   35.3  !
! $255$DJA9                  1.9     0.3     0.0    11.3     0.3   26.3    39.2     65      67    0.0   83.5  !
```

Figure 7–3 Tabular Report for Diagnosing a CPU Limitation

Figure 7–4 Scheduler States Section of Tabular Report

```
+-------------------------------------------------------- Scheduler States -----------------------------------------------------+
!                                                                                                                               !
!                                                      Real Time Priorities                                                     !
!                                                      --------------------                                                     !
!      State        Total       31     30     29     28     27     26     25     24     23     22     21     20     19     18     17     16  !
!                                                                                                                               !
!  HIB                2         -      -      -      -      -      -      -      -      -      -      -      -      -      -      -      2   !
!  CUR                1         -      -      -      -      -      -      -      1      -      -      -      -      -      -      -      -   !
!  COM                1         -      -      -      -      -      -      -      1      -      -      -      -      -      -      -      -   !
!                                                                                                                               !
!                                                     Timesharing Priorities                                                    !
!                                                     ----------------------                                                    !
!      State        Total       15     14     13     12     11     10      9      8      7      6      5      4      3      2      1      0   !
!                                                                                                                               !
!  LEF               13         -      -      -      -      -      4      2      1      1      2      1      2      -      -      -      -   !
!  HIB               22         -      3      -      1      -      1      2      4      2      3      1      5      -      -      -      -   !
+-------------------------------------------------------------------------------------------------------------------------------+
```

Table 7–5 Diagnosing a CPU Limitation

Rule	Probable Cause
AVE CPU Queue > 1	Too Many Processes in CPU Queue
Scheduler States Com > 1	Too Many Processes in CPU Queue
Total Idle ≤ 15%	No Idle Time
System > Task and Task > 25% of the total CPU time	High System CPU Time

7.4 Chapter Summary

There are several steps in the tuning methodology: diagnose, isolate, and correct the problem, and verify the results of your correction. This chapter explains the diagnosis phase of tuning.

The following command generates three reports needed to diagnose a limiting resource: the Final Tabular, System Parameters, and Scheduler States reports.

```
$ PERFORMANCE REPORT=LOG_FILE/TABULAR=FINAL -
/CLASS=(SCHEDULER,SYSGEN_PARAMETERS) -
/BEFORE=beginning-time of hot spot/SINCE=end-time of hot spot -
SPM$COLLECT_TUNE.DAT
```

Table 7–3, Table 7–4, and Table 7–5 contain rules to diagnose memory, I/O, and CPU limitations, respectively. Any resource for which a rule is true may be a limiting resource. Once you have diagnosed a limiting resource, you are ready to isolate the cause as described in Chapter 8.

The two types of tabular report are interval and final. Although Final Tabular reports are most often used in the tuning methodology, Interval reports are useful in investigating excessive image activations and hot spots with varying levels of activity. The following command generates Interval Tabular reports:

```
$ PERFORMANCE REPORT=LOG_FILE/TABULAR=INTERVAL -
/BEFORE=beginning-time of extreme activity within hot spot -
/SINCE=end-time of extreme activity within hot spot -
SPM$COLLECT_TUNE.DAT
```

8 Isolating the Resource Limitation

This chapter describes isolating the resource that is limiting system performance by using VAX SPM reports, and provides the following information:

- Introduction to the process of isolating limiting resources
- Instructions for isolating the cause of a memory limitation
- Instructions for isolating the cause of an I/O limitation
- Instructions for isolating the cause of a CPU limitation

8.1 Introduction

In Chapter 7 you used the Final Tabular, System Parameters, and Scheduler States reports to diagnose a limiting resource on your system. In this chapter, you will isolate the cause of the limitations using the Final Tabular report and two additional reports: the Process Metric and Extended Process Metric reports. These reports contain detailed information about each process. See the *VAX SPM Reference Manual* for descriptions of these reports.

Use the following command to generate Interval Tabular reports and Interval Process Metrics reports that include standard process metrics, extended process metrics, and image names.

```
$ PERFORMANCE REPORT=LOG_FILE/CLASS=(PROCESS:ALL)/TABULAR=INTERVAL -
/NOGRAPH/BEFORE=beginning-time-of-hot-spot/SINCE=end-time-of-hot-spot -
SPM$COLLECT_TUNE.DAT
```

The procedure for isolating a resource limitation is similar to the procedure you used for diagnosing a resource limitation. The following sections are parallel to the discussion in Chapter 4 of the *Guide to VAX/VMS Performance Management* and describe how to evaluate VAX SPM reports to isolate resource limitations. Examine your reports as described and when you have determined the cause, apply the approaches mentioned here and discussed in the *Guide to VAX/VMS Performance Management* to correct the problem.

Find the resource you diagnosed as limiting in Chapter 7 in the following list and turn to the corresponding section to isolate the cause of that limitation.

- Memory as the limiting resource—Section 8.2, Isolating a Memory Limitation
- I/O as the limiting resource—Section 8.3, Isolating an I/O Limitation
- CPU as the limiting resource—Section 8.4, Isolating a CPU Limitation

8.2 Isolating a Memory Limitation

Examine your reports as described in each of the following sections to isolate the cause of a memory limitation.

8.2.1 High Page Faulting

Look at the tabular report field labeled Page Faults. If this rate is above 100 times the CPU factor for your system, page faulting may be excessive. (See Table 7–2 for the CPU factor associated with your system.)

There are two types of paging: hard paging (from disk) and soft paging (from page caches in main memory). Hard paging causes I/O to disk and is less desirable than soft paging. Soft paging is less costly to system performance because it does not require disk I/O.

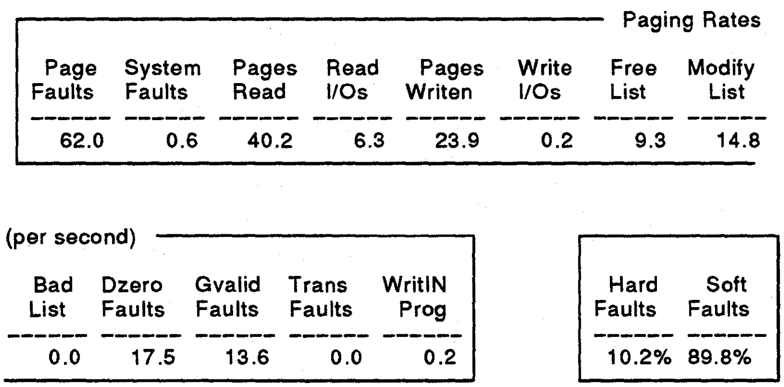
Table 8–1 describes the composition of the Hard Fault field and Soft Fault fields in the tabular report:

Table 8–1 Composition of Fault Fields in Tabular Report

Hard Faults	Soft Faults	
Read I/Os	Free List	Gvalid
	Modify List	Trans
	Bad List	WritIN Prog
	Dzero	

Figure 8–1 displays the Paging Rates section of the tabular report.

Figure 8–1 Paging Rates Section of Tabular Report



```
 ------------------------------------------------ Paging Rates
|
|  Page  System   Pages   Read    Pages   Write   Free  Modify
| Faults Faults   Read    I/Os   Writen   I/Os    List    List
| ------ ------ ------ ------ ------ ------ ------ ------
|   62.0    0.6   40.2    6.3   23.9    0.2    9.3   14.8
 ------------------------------------------------------------

(per second) ---------------------------            ----------------
                                        |          |                |
  Bad  Dzero   Gvalid  Trans   WritIN   |          |  Hard    Soft  |
  List Faults  Faults  Faults    Prog   |          | Faults  Faults |
------ ------ ------ ------ ------      |          | ------ ------  |
   0.0   17.5   13.6    0.0    0.2      |          |  10.2% 89.8%   |
-----------------------------------------           ----------------
```

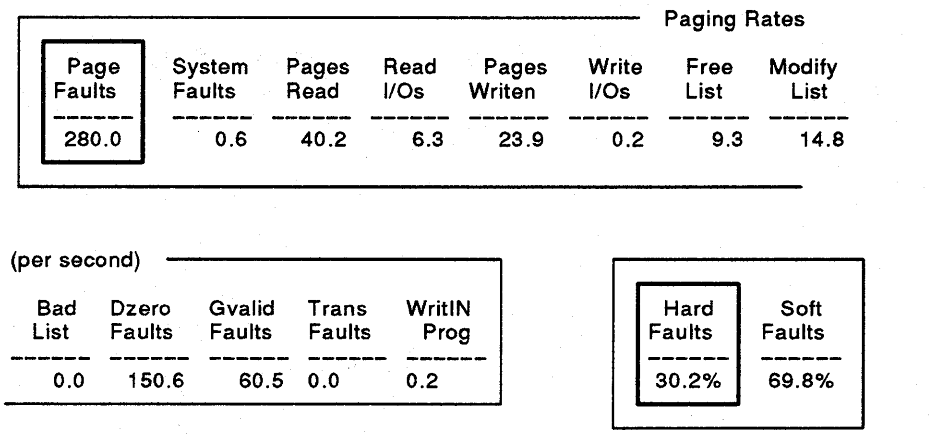
If your system does not have a high page fault rate, proceed to Section 8.2.2. Otherwise, examine your reports as described in the following sections to further isolate the causes of high page faulting.

8.2.1.1 Too Many Image Activations

Look at the Dzero Fault and Page Fault fields in the Paging Rates section of the tabular report. A value of Dzero Fault greater than or equal to 50% of all Page Faults may indicate too many image activations. For example, in Figure 8–2 the value of Dzero Faults (150.6) is greater than 50% of the value of Page Faults (280).

In addition, a Hard Fault rate greater than 10% of the Page Fault rate may indicate too many image activations. In Figure 8–2, the Hard Fault rate (30) is greater than 10% of the Page Faults (280), possibly indicating excessive image activations.

Figure 8–2 Tabular Report Segment for Isolating Excessive Image Activations



```
                                                      Paging Rates
 Page     System  Pages   Read    Pages   Write   Free    Modify
 Faults   Faults  Read    I/Os    Writen  I/Os    List     List
 ------   ------  ------  ------  ------  ------  ------  ------
 280.0      0.6    40.2     6.3    23.9     0.2     9.3    14.8

(per second)

 Bad     Dzero   Gvalid  Trans   WritIN              Hard     Soft
 List    Faults  Faults  Faults   Prog               Faults   Faults
 ------  ------  ------  ------  ------              ------   ------
  0.0    150.6    60.5   0.0     0.2                 30.2%    69.8%
```

Once you have identified excessive image activations, the next step is to determine the processes responsible for the excessive image activations. Look at the Image Count column in the Process Metrics report. This column lists the number of image activations for each process. For example, in Figure 8–3 B_USER has 16 image activations per interval.

Once you have determined which process is responsible for the image activations, find out which images are activated. Select a time of day based upon previous days' reports during which excessive image activations usually occur. Using a small sample interval (for example, 10 to 30 seconds), collect tuning data for this time period and generate Process Metrics Interval reports. The Process Metrics report shows the names of the images that are activated.

Figure 8–3 Process Metrics Report Showing Image Count Column

```
 --------------- Process Metrics ---------------
                            \\
           Process          //    Image     CPUtime
   PID     Name             \\    Count       (min)
 -------- ------------------//   --------   ------
 00000100 A_USER            \\        0     0.025
 00000101 B_USER            //       16     1.020
          ------            \\       --
 00000104 C_USER            //        8      .455
 00000105 D_USER            \\        2      .099
```

Figure 8–4 Process with Rapidly Changing Image Name

```
SDML data collected from 26-OCT-1984 09:15:27 to 26-OCT-1984 09:15:57

                                 \\                  \\  Page Flts    Working Set
    Process ID  Process Name     //  CPUtime(min)    //  /sec         Globl/ Priv
    ---------- ------------      \\  -----------     \\  ---------    ----- -----
      20200220 V54:B_USER        //         0.230    //     30.340        7   251
            $255$DJS8:[B_USER]ATST.EXE;1

SDML data collected from 26-OCT-1984 09:15:57 to 26-OCT-1984 09:16:27

                                 \\                  \\   Page Flts   Working Set
    Process ID  Process Name     // CPUtime(min)     //   /sec        Globl/ Priv
    ---------- ------------      \\ ------------     \\ ----------    ----- -----
      20200220 V54:B_USER        //         0.161    //     25.010        7   180
            $255$DJS8:[B_USER]DRAW.EXE;1

SDML data collected from 26-OCT-1984 09:16:27 to 26-OCT-1984 09:16:57

                                 \\                  \\   Page Flts   Working Set
    Process ID  Process Name     // CPUtime(min)     //   /sec        Globl/ Priv
    ---------- ------------      \\ ------------     \\ --------      ----- -----
      20200220 V54:B_USER        //         0.192    //  39.749           7   260
           $255$DJS8:[B_USER]DOCUMENT.EXE;1
```

Do not collect data at this small sample interval rate for more than an hour because this creates a rather large log file. If you have limited disk space on your system, refer to the *VAX SPM Reference Manual* to estimate the size of a log file and adjust your collection time and sample interval accordingly.

Examine several consecutive pages of the Process Metrics report, looking for processes whose image name is changing rapidly. These processes may cause excessive page faults. Figure 8–4 is an example.

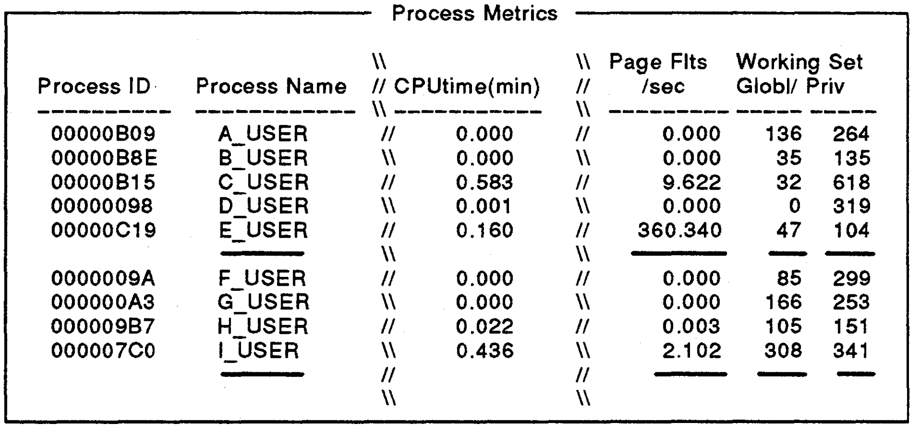
8.2.1.2 Inappropriate Automatic Working Set Sizes

The most common cause of page faulting problems is inappropriate working set sizes. The system allocates working set sizes according to the values of the user authorization file (UAF) parameters WSDEFAULT, WSQUOTA, and WSEXTENT. If your system is short on memory, redistribute the working set sizes so that active processes get more memory. The *VMS Authorize Utility Manual* describes how to change the value of UAF parameters.

Look at the Process Metrics and Extended Process Metrics reports. Processes with inappropriate working set sizes may be distinguished in two ways: by a small working set size with a high page fault rate or by a large working set size with a low page fault rate.

For example, in Figure 8–5 process E_USER has a page fault rate per second of 360, a global working set size of 47, and a private working set size of 104. Process I_USER has a page fault rate of 2 per second, a global working set of 308 pages, and a private working set size of 341 pages. Both processes exhibit symptoms of inappropriate working set sizes.

Figure 8–5 Process Metrics for Inappropriate WS Size



```
 ------------------------------- Process Metrics --------------------------------
                                  \\                  \\  Page Flts   Working Set
  Process ID   Process Name      // CPUtime(min)      //    /sec      Globl/ Priv
  ----------   ------------      \\ ----------        \\ ---------   ----- -----
   00000B09      A_USER          //       0.000       //     0.000     136   264
   00000B8E      B_USER          \\       0.000       \\     0.000      35   135
   00000B15      C_USER          //       0.583       //     9.622      32   618
   00000098      D_USER          \\       0.001       \\     0.000       0   319
   00000C19      E_USER          //       0.160       //   360.340      47   104
                 ------          \\                   \\   -------     ---   ---
   0000009A      F_USER          //       0.000       //     0.000      85   299
   000000A3      G_USER          \\       0.000       \\     0.000     166   253
   000009B7      H_USER          //       0.022       //     0.003     105   151
   000007C0      I_USER          \\       0.436       \\     2.102     308   341
                 ------          //                   //   -------     ---   ---
                                 \\                   \\
```

Look for processes that fault heavily but have small working sets. These processes have working sets that are too small for their workload. You may decide to increase WSQUOTA and WSEXTENT for these processes.

Look for processes that have large working sets but are not particularly active (low page fault rate). These processes have more memory than they need. Reducing WSQUOTA and/or WSEXTENT for these processes allows other processes to use the memory.

If your system has ample memory, increase the working set parameter values for heavily faulting processes.

8.2.1.3 Inappropriate Working Set Adjustment Parameters

The rapid fluctuation of process working set sizes from interval to interval may indicate that working set adjustment parameters may have inappropriate values.

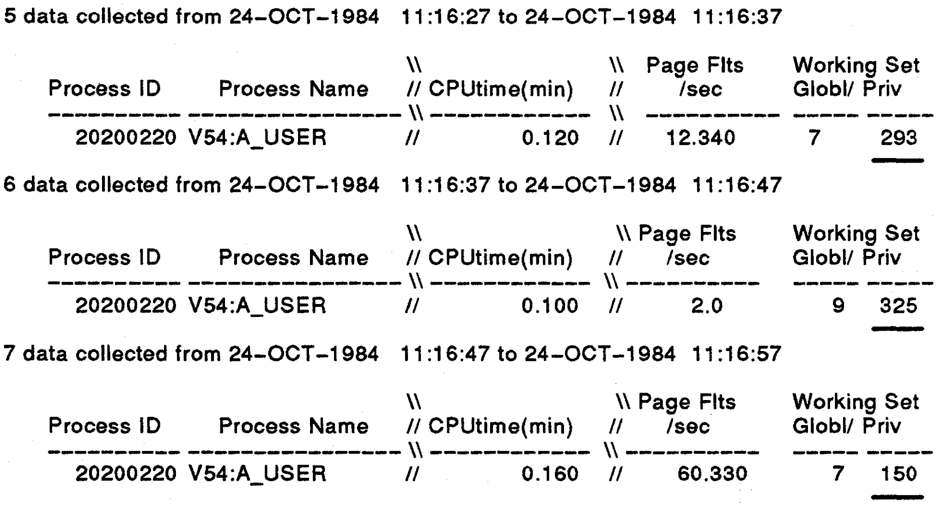
From previous days' reports, select a time when high page faulting is most likely to occur. For this time, collect tuning data using a small sample interval (for example, 10 to 30 seconds).

Do not collect more than one hour's data because the log file will get very large. If you have limited disk space on your system, refer to the *VAX SPM Reference Manual* to learn how to estimate the size of a log file, and adjust your collection time and sample interval accordingly.

Generate Interval Process Metrics reports. Examine the reports, looking for processes whose working set sizes fluctuate rapidly.

In Figure 8–6, the private working set size for process A_USER grew from 293 to 325 pages, then went down immediately to 150, close to the value of the system parameter SWPOUTPGCNT at 157 pages. Notice that the page faults per second vary in the same manner that working set sizes vary.

Figure 8–6 Process Metric Reports Showing Fluctuating Working Set Sizes



```
5 data collected from 24-OCT-1984  11:16:27 to 24-OCT-1984  11:16:37

                                  \\                 \\  Page Flts    Working Set
   Process ID    Process Name     // CPUtime(min)    //    /sec       Globl/ Priv
   ---------- ------------------ \\ ------------- \\  ----------   ----- -----
     20200220 V54:A_USER          //         0.120   //   12.340       7    293

6 data collected from 24-OCT-1984  11:16:37 to 24-OCT-1984  11:16:47

                                  \\                 \\ Page Flts     Working Set
   Process ID    Process Name     // CPUtime(min)    //    /sec       Globl/ Priv
   ---------- ------------------ \\ ------------- \\ ----------     ----- -----
     20200220 V54:A_USER          //         0.100   //     2.0          9   325

7 data collected from 24-OCT-1984  11:16:47 to 24-OCT-1984  11:16:57

                                  \\                 \\ Page Flts     Working Set
   Process ID    Process Name     // CPUtime(min)    //    /sec       Globl/ Priv
   ---------- ------------------ \\ ------------- \\ ----------     ----- -----
     20200220 V54:A_USER          //         0.160   //    60.330        7   150
```

System default values for automatic working set adjustment (AWSA) parameters as set by AUTOGEN are normally adequate; do not modify key parameter values without a thorough understanding of the entire automatic working set adjustment mechanism. In addition, the *VAX/VMS System Generation Utility Manual* provides information about system parameters, their optimal values, and how to change them when necessary.

If AWSA parameters are out of adjustment, excessive paging can result. The following list describes some AWSA problems and some recommendations for resolving them:

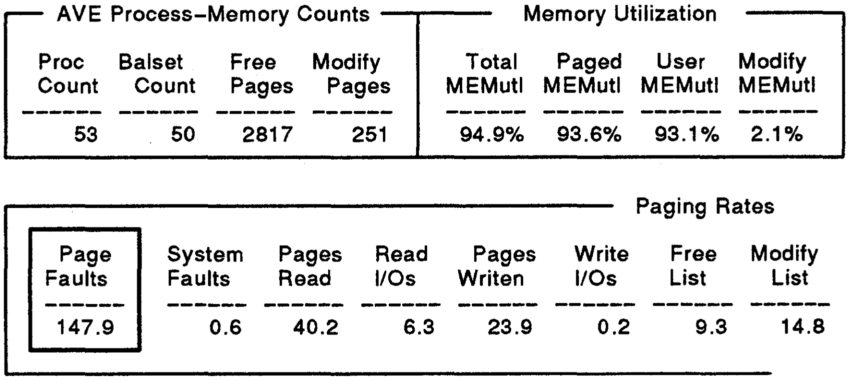
- AWSA may not be increasing working set sizes quickly enough to respond to faulting. Adjust WSINC, PFRATH, and AWSTIME, and consider adjusting WSDEFAULT and WSQUOTA of the offending processes.
- The voluntary decrementing feature of AWSA may cause oscillation in which working set sizes never stabilize but grow and shrink. This oscillation is accompanied by page faulting. Turn off voluntary decrementing by setting PFRATL=0.
- AWSA shrinks working set sizes too quickly. Decrease WSDEC and/or PFRATL.
- Processes achieve very large working set sizes and maintain those sizes. They do not page fault but other processes do. Turn on voluntary decrementing by setting PFRATL to nonzero. This is the exception rather than the rule.

8.2.1.4 Inappropriate Memory Loan Parameters

Look at the tabular report fields Free Pages and Page Faults. If there is enough free memory but a high number of page faults, memory loan parameters may be inappropriate.

Figure 8–7 is an example of a tabular report displaying symptoms of inappropriate values for memory loan parameters. The tabular report shows an ample amount of free memory (Free Pages has a value of 2817) and a high number of Page Faults (147). This report is for a VAX-11/750 system for which a page faulting rate above 65 is considered high. (Table 7–2 shows a CPU Factor of .65 for a VAX-11/750. This figure is multiplied by 100 as described in Table 7–3 to determine that a page fault above 65 is high for a VAX-11/750.)

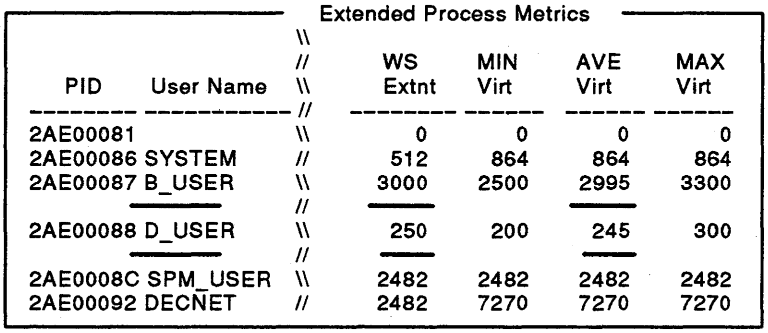
Figure 8–7 Free Memory + High Page Faults



```
  AVE Process-Memory Counts              Memory Utilization

 Proc   Balset    Free   Modify     Total   Paged   User   Modify
 Count   Count    Pages   Pages    MEMutl  MEMutl  MEMutl  MEMutl
------  ------   ------  ------    ------  ------  ------  ------
   53      50     2817     251      94.9%   93.6%   93.1%   2.1%

                                                  Paging Rates

  Page     System   Pages   Read    Pages   Write   Free   Modify
 Faults    Faults   Read    I/Os    Writen  I/Os    List    List
 ------    ------  ------  ------  ------  ------  ------  ------
  147.9      0.6    40.2     6.3    23.9     0.2     9.3    14.8
```

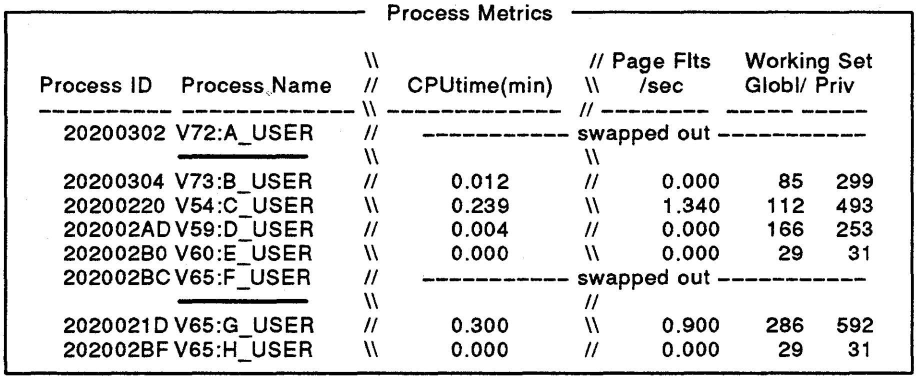
Another symptom of inappropriate memory loan parameters may be observed in the Extended Process Metrics report. Look for a lot of processes that have their working set sizes approaching their WSEXTENT values while other processes are swapped out. For example, in Figure 8–8 process B_USER has a WS Extent value of 3000 and an AVE Virt working set size of 2995, and D_USER has a WS Extent value of 250 and an AVE Virt working set size of 245. Examining the Process Metrics reports for this interval reveals that processes A_USER and F_USER are both swapped out as shown in Figure 8–9.

Figure 8–8 Extended Process Metrics Report Showing Inappropriate Memory Loan Parameters



```
                               Extended Process Metrics
                             \\
                             //      WS     MIN      AVE     MAX
   PID      User Name        \\      Extnt  Virt     Virt    Virt
 --------- -----------       //     ------- ------  ------ ------
 2AE00081                    \\          0      0       0       0
 2AE00086 SYSTEM             //        512    864     864     864
 2AE00087 B_USER             \\       3000   2500    2995    3300
                             //
 2AE00088 D_USER             \\        250    200     245     300
                             //
 2AE0008C SPM_USER           \\       2482   2482    2482    2482
 2AE00092 DECNET             //       2482   7270    7270    7270
```

Figure 8–9 Example of a Process Metrics Report Showing Inappropriate Memory Loan Parameters



```
                                  Process Metrics

                             \\                    // Page Flts   Working Set
 Process ID   Process Name   //  CPUtime(min)    \\    /sec      Globl/ Priv
 ----------  --------------  \\  -------------   // --------   ----- -----
   20200302 V72:A_USER       //   ----------- swapped out -----------
                             \\                  \\
   20200304 V73:B_USER       //      0.012       //    0.000       85   299
   20200220 V54:C_USER       \\      0.239       \\    1.340      112   493
   202002AD V59:D_USER       //      0.004       //    0.000      166   253
   202002B0 V60:E_USER       \\      0.000       \\    0.000       29    31
   202002BC V65:F_USER       //   ----------- swapped out -----------
                             \\                  //
   2020021D V65:G_USER       //      0.300       \\    0.900      286   592
   202002BF V65:H_USER       \\      0.000       //    0.000       29    31
```

The following are possible causes of the symptoms that indicate inappropriate values for memory loan parameters:

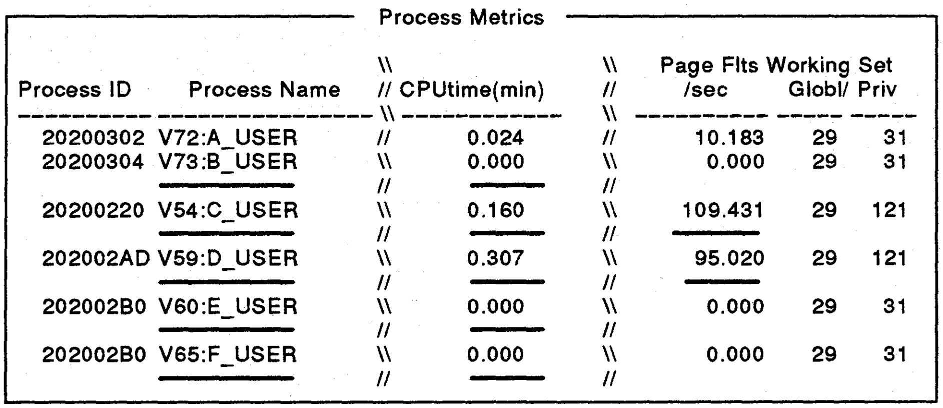
- BORROWLIM is too high and the system is not loaning memory.
- PFRATH may be too high, forcing processes into high fault rates before they are given additional memory.
- WSEXTENT may be too low so that processes reach their limit and are still faulting heavily.
- If BORROWLIM or GROWLIM are too generous (set too low), memory may be given away inappropriately.

8.2.1.5 Inappropriate Swapper Trimming

Look at the Process Metrics and Extended Process Metrics reports, especially the active processes (top faulting processes with some CPU time). If many active processes have working set sizes at their WSQUOTA value or that of SWPOUTPGCNT, idle processes are not getting swapped and active processes are getting trimmed.

For example, in Figure 8–10 processes C_USER and D_USER are active processes (with 0.160 and 0.30 CPU time and 109 and 95 page faults per second, respectively) while processes B_USER, E_USER, and F_USER are inactive and not swapped out.

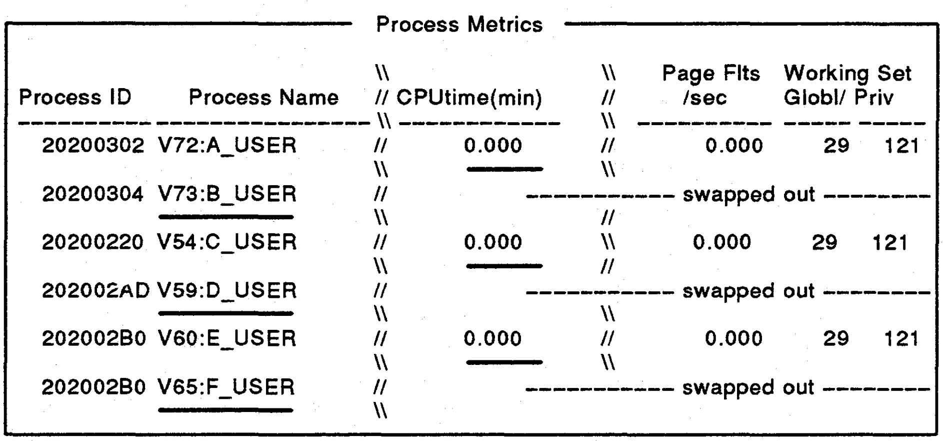
Figure 8–10 Process Metrics Report Showing Inappropriate Swapper Trimming



```
                              Process Metrics

                                 \\                  \\   Page Flts Working Set
Process ID     Process Name      // CPUtime(min)     //     /sec     Globl/ Priv
---------- ----------------- \\ ------------ \\ ---------- ----- -----
  20200302 V72:A_USER          //      0.024       //      10.183     29    31
  20200304 V73:B_USER          \\      0.000       \\       0.000     29    31
                               //                  //
  20200220 V54:C_USER          \\      0.160       \\     109.431     29   121
                               //                  //
  202002AD V59:D_USER          \\      0.307       \\      95.020     29   121
                               //                  //
  202002B0 V60:E_USER          \\      0.000       \\       0.000     29    31
                               //                  //
  202002B0 V65:F_USER          \\      0.000       \\       0.000     29    31
                               //                  //
```

Another symptom of inappropriate swapper trimming occurs when the value of SWPOUTPGCNT is too large. When this happens, some processes get swapped out while other processes use little or no CPU time. For example, in Figure 8–11 processes B_USER, D_USER and F_USER are swapped out while processes A_USER, C_USER, and E_USER are not using any CPU time.

Figure 8–11 Process Metrics Report Showing SWPOUTPGCNT Too Large

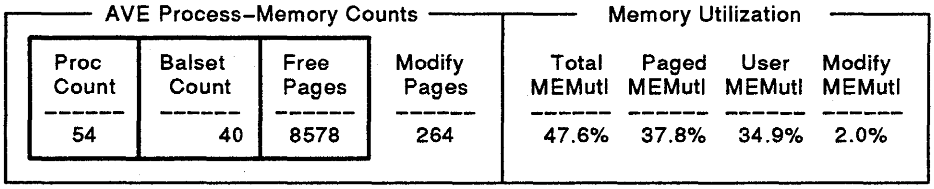


```
------------------------------ Process Metrics -----------------------------
                                 \\                \\   Page Flts  Working Set
Process ID     Process Name      // CPUtime(min)   //     /sec     Globl/ Priv
-------- ----------------------- \\ -----------    \\ ---------- ----- -----
  20200302  V72:A_USER           //      0.000     //     0.000       29   121
                                 \\                \\
  20200304  V73:B_USER           //           ----------- swapped out --------
                                 \\                //
  20200220  V54:C_USER           //      0.000     \\    0.000       29   121
                                 \\                //
  202002AD V59:D_USER            //           ----------- swapped out --------
                                 \\                \\
  202002B0  V60:E_USER           //      0.000     //     0.000       29   121
                                 \\                \\
  202002B0  V65:F_USER           //           ----------- swapped out --------
                                 \\
```

8.2.2 Balance Set Count Too Small

Look at Proc Count, Balset Count, and Free Pages fields on the tabular report. If Balset Count is significantly less than Proc Count, and the value of Free Pages is adequate, then the balance set count is too low. Figure 8–12 is an example of a tabular report showing Balset Count with a value of 40, which is significantly less than Proc Count whose value is 54 while there are 8578 Free Pages.

Figure 8–12 Tabular Report Segment Showing Balance Set Count Too Small



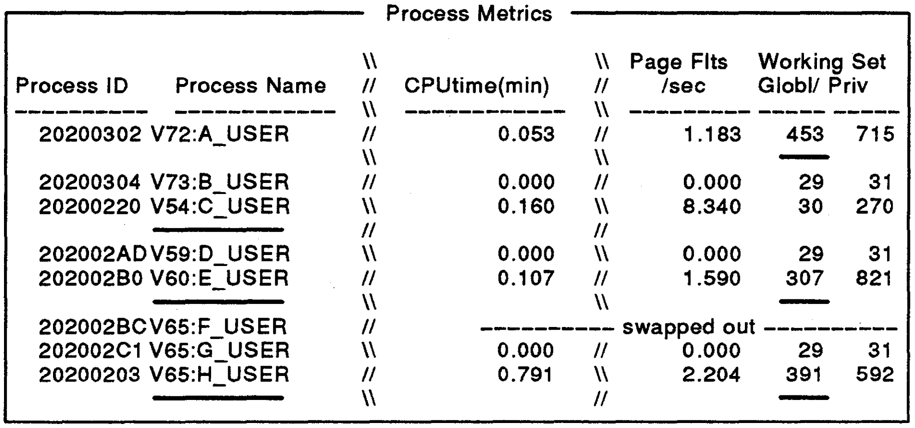
```
+------- AVE Process-Memory Counts -------+------- Memory Utilization -------+
|  Proc    Balset    Free     Modify      |  Total   Paged   User    Modify   |
|  Count   Count     Pages    Pages       |  MEMutl  MEMutl  MEMutl  MEMutl   |
|  ------  ------    ------   ------      |  ------  ------  ------  ------   |
|   54         40    8578      264        |  47.6%   37.8%   34.9%   2.0%     |
+-----------------------------------------+-----------------------------------+
```

8.2.3 A Few Active Processes Consuming Memory

A few large compute-bound processes may consume memory at the expense of a number of smaller processes. A few processes with extremely large working sets (working set sizes approaching the value of WSEXTENT when they should not be) may cause other processes to swap. For example, a low priority compute-bound process is less likely to be swapped than one that performs terminal I/O. When the latter process goes into terminal I/O wait, it may be outswapped.

In Figure 8–13 processes C_USER, E_USER, and H_USER have large working set sizes and are using significant amounts of CPU time, while process F_USER is swapped out.

Figure 8–13 Compute-Bound Processes Consuming Memory



```
---------------------------- Process Metrics -----------------------------
                                 \\                 \\  Page Flts  Working Set
Process ID    Process Name       //  CPUtime(min)   //    /sec     Globl/ Priv
----------   -------------       \\  ------------   \\  --------- ----- -----
  20200302 V72:A_USER            //         0.053   //     1.183    453   715
                                 \\                 \\              ---
  20200304 V73:B_USER            //         0.000   //     0.000     29    31
  20200220 V54:C_USER            \\         0.160   \\     8.340     30   270
           ------------          //                 //
  202002ADV59:D_USER             \\         0.000   \\     0.000     29    31
  202002B0 V60:E_USER            //         0.107   //     1.590    307   821
           ------------          \\                 \\              ---
  202002BCV65:F_USER             //      ----------- swapped out -----------
  202002C1 V65:G_USER            \\         0.000   //     0.000     29    31
  20200203 V65:H_USER            //         0.791   \\     2.204    391   592
           ------------          \\                 //              ---
```

Decreasing DORMANTWAIT may help provide more memory for smaller processes by causing large processes to swap out if the large processes are above their working set quota.

You can also suspend the large process with the DCL command SET PROCESS/SUSPEND. This allows the swapper to trim the process to its SWPOUTPGCNT value, freeing memory for other processes. As long as memory is tight, automatic working set adjustment will prevent the compute-bound process from growing beyond its WSQUOTA again.

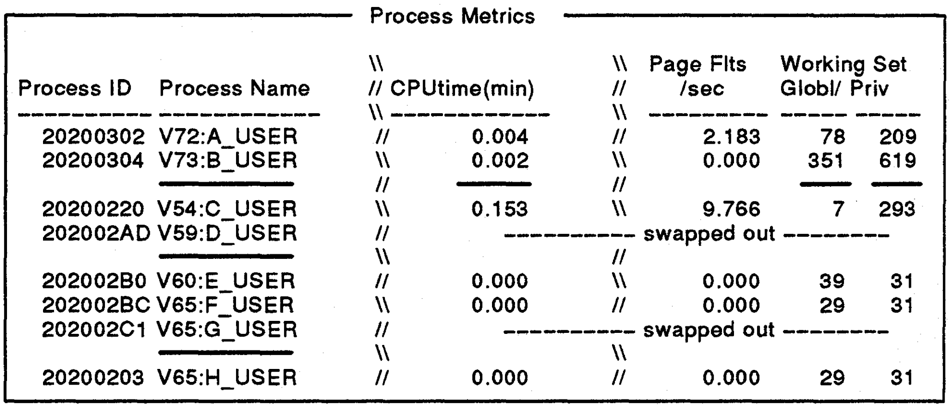
Whatever action you take, examine the underlying cause of the problem. For example, WSQUOTA may be too large for the process.

8.2.4 Large Process with Swapping Disabled

Inactive processes (no page faulting) with large working sets may have swapping disabled. This means they cannot be swapped out, and retain memory at the expense of other processes.

Look at the Process Metrics report for inactive processes with large working sets. For example, in Figure 8–14 process B_USER is using little or no CPU time and is inactive (indicated by a lack of page faults). Processes D_USER and G_USER are swapped out. Process B_USER has a working set of 351 global and 619 private pages, which could be made available to active processes if it were swapped out. The next step is to see if process B_USER has swapping disabled.

Figure 8–14 Large Processes with Swapping Disabled



```
                                Process Metrics
                              \\                   \\  Page Flts   Working Set
Process ID  Process Name      // CPUtime(min)      //    /sec      Globl/ Priv
----------  ------------      \\ -------------     \\ ---------- ----- -----
  20200302  V72:A_USER        //       0.004       //     2.183     78   209
  20200304  V73:B_USER        \\       0.002       \\     0.000    351   619
            ----------        //     -------       //               ----  ----
  20200220  V54:C_USER        \\       0.153       \\     9.766      7   293
  202002AD  V59:D_USER        //          ----------- swapped out --------
            ----------        \\                   //
  202002B0  V60:E_USER        //       0.000       \\     0.000     39    31
  202002BC  V65:F_USER        \\       0.000       //     0.000     29    31
  202002C1  V65:G_USER        //          ----------- swapped out ---------
            ----------        \\                   \\
  20200203  V65:H_USER        //       0.000       //     0.000     29    31
```

You can invoke the system dump analyzer (SDA) with the DCL command ANALYZE/SYSTEM to determine if swapping is disabled. Use the SDA command SHOW PROCESS to see if any process status line has a status of PSWAPM (prohibit swap mode).

```
$ ANALYZE/SYSTEM

VMS System analyzer

SDA> SHOW SUMMARY

Current process summary
-----------------------
Extended Indx Process name    Username    State Pri   PCB      PHD     Wkset
- PID -- ---- --------------- ----------- ----- --- -------- -------- -----
20400201 0001 SWAPPER                      HIB   16 80002748 800025C8     0
20400206 0006 ERRFMT          SYSTEM       HIB    8 8042EB40 80924400   115
2040046E 006E SPM_USER        SPM          LEF   24 80468C90 81AF3C00   468
20400472 0072 V71:A_USER      A_USER       LEF    8 80468B60 81955400  1080
20400673 0073 RA3:B_USER      B_USER       CUR    6 8046FFC0 839B7400   876
20400474 0074 V72:C_USER      C_USER       LEF    4 80468280 81886000   759
20400477 0077 V74:D_USER      D_USER       LEF    8 8046A260 81BC3000   266

SDA> SET PROCESS/INDEX=006E
SDA> SHOW PROCESS

Process index: 006E   Name: SPM_USER    Extended PID: 2040046E
----------------------------------------------------------------
Process status:  00040011   RES,PSWAPM,PHDRES
```

If a process has swapping disabled, consider whether it should be made swappable. A process should have swapping disabled in the following cases:

- Most real-time processes should not be swapped.
- The owner of the process may have valid reasons why the process should not be swapped.

If the owner of the process agrees, enable the process for swapping with DCL command SET PROCESS/SWAPPING (requires PSWAPM privilege).

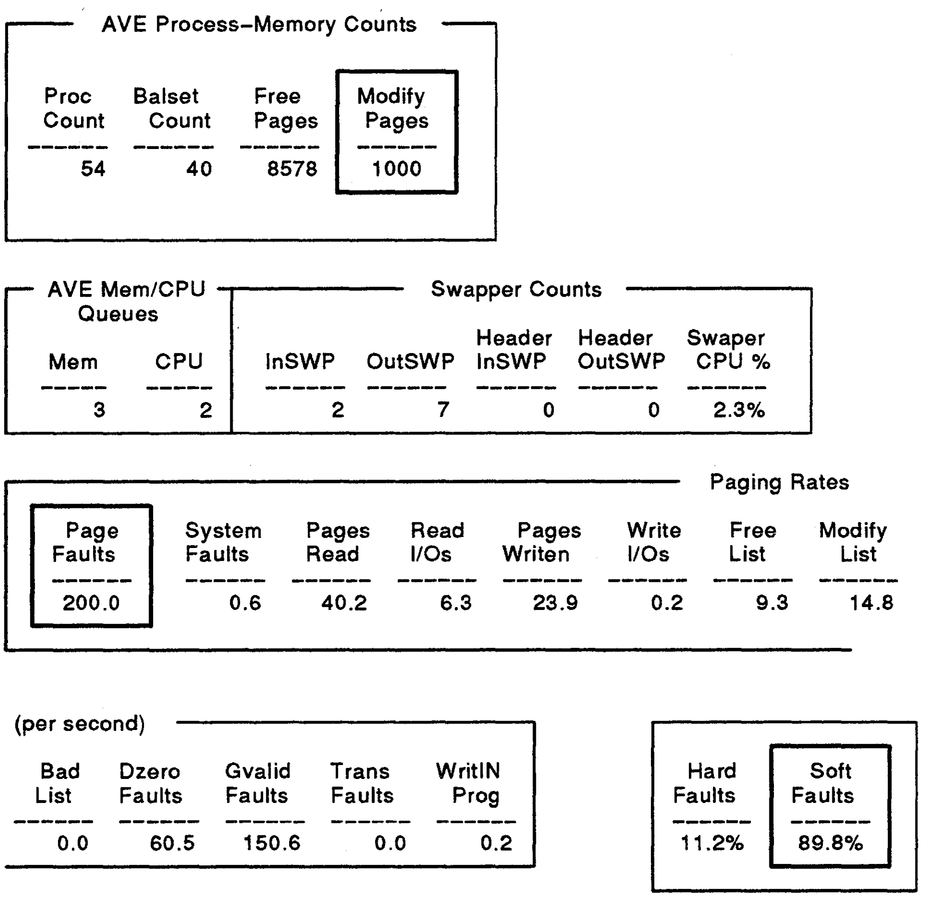
If the offending process is a disk ACP (ODS-1 only), set the system parameter ACP_SWAPFLGS appropriately and reboot the system. See the *VAX/VMS System Generation Utility Manual* for further information.

8.2.5 Inappropriate Page Cache Sizes

Page Cache Too Large

If the overall fault rate is high and the faults are mostly soft faults, the page cache may be too large. This may also be accompanied by swapping and extensive free and modified page lists. The page cache is using memory that could be made available for working sets. For example, too large a page cache may be indicated in Figure 8–15. The Page Fault rate is 200 and the Soft Fault rate is 89%, accompanied by a Modified Page list of 1000 pages.

Figure 8–15 Tabular Report Segment for Determining Too Large a Page Cache



```
      AVE Process-Memory Counts

  Proc   Balset    Free   Modify
  Count   Count   Pages    Pages
 ------  ------  ------   ------
     54      40    8578    1000


 AVE Mem/CPU               Swapper Counts
   Queues
                                 Header  Header   Swaper
  Mem     CPU    InSWP  OutSWP   InSWP   OutSWP   CPU %
 -----   -----   ------ ------  ------  ------   ------
     3       2        2      7       0       0     2.3%

                                                        Paging Rates
   Page     System   Pages   Read    Pages    Write   Free    Modify
  Faults    Faults   Read    I/Os    Writen   I/Os    List     List
  ------    ------  ------  ------   ------  ------  ------   ------
  200.0        0.6    40.2     6.3     23.9     0.2     9.3     14.8


 (per second)
   Bad    Dzero   Gvalid   Trans   WritIN           Hard    Soft
   List   Faults  Faults   Faults   Prog           Faults  Faults
 ------  ------  ------   ------  ------          ------  ------
    0.0    60.5   150.6      0.0     0.2           11.2%   89.8%
```

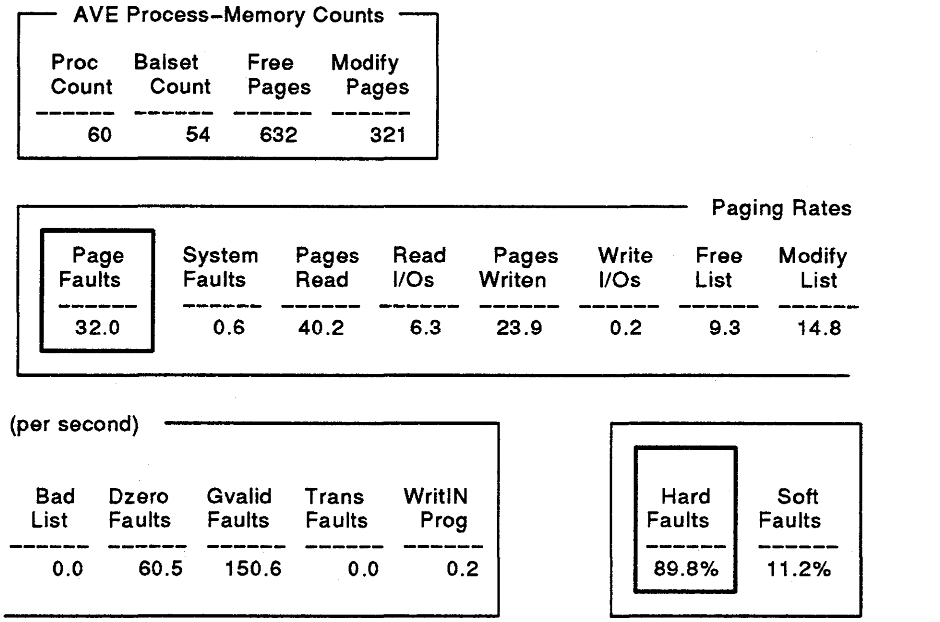
Page Cache Too Small

If the overall faulting rate is low while the hard fault rate is high, the page cache may be ineffective; that is, the free page list and/or modified page list is too small. There is ample memory for working sets but the caching effectiveness is low.

For example, in Figure 8–16 the Page Fault rate is low at 32 per second, while Hard Faults account for 89% of all page faults. Values for the Free Pages and Modify Pages fields may indicate that these lists are small; therefore, the page cache may be too small.

The sizes of the page caches are controlled by the system parameters FREELIM, FREEGOAL, MPW_LOLIMIT, and MPW_THRESH. You may need to modify these parameters to achieve appropriate page cache sizes. The objective is to balance the page cache size against user memory.

Figure 8–16 Tabular Report Segment Indicating Too Small a Page Cache



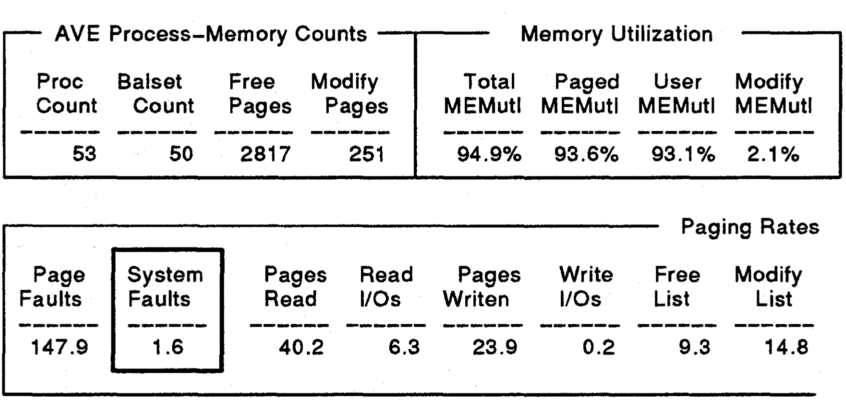
```
┌─ AVE Process-Memory Counts ─┐
 Proc   Balset   Free   Modify
 Count   Count   Pages   Pages
 ------ ------ ------ ------
    60      54    632     321

┌─────────────────────────────────────────────────── Paging Rates
 Page     System  Pages   Read    Pages   Write   Free   Modify
 Faults   Faults  Read    I/Os    Writen  I/Os    List    List
 ------   ------ ------ ------ ------ ------ ------ ------
  32.0       0.6   40.2    6.3    23.9    0.2    9.3    14.8

(per second) ─────────────────────

 Bad    Dzero   Gvalid  Trans   WritIN        Hard     Soft
 List   Faults  Faults  Faults    Prog        Faults   Faults
 ------ ------ ------ ------ ------          ------   ------
   0.0    60.5  150.6     0.0     0.2          89.8%    11.2%
```

8.2.6 System Working Set Too Small

Look at the tabular report field System Faults (under Paging Rates). If this value is greater than 1.0, the current system working set size may be too small. For example, in Figure 8–17 the System Faults field has a value of 1.6, which may indicate that the system working set is too small.

The presence of system page faults may indicate that the system needs a larger working set. If there is adequate free memory, increase the system parameter SWSMWCNT by 50-page increments until the system fault rate is 1.0 or less. You must reboot the system before this change will take effect.

Figure 8–17 Tabular Report Segment Indicating Insufficient System Working Set



```
 AVE Process-Memory Counts              Memory Utilization
 Proc   Balset   Free   Modify     Total   Paged   User   Modify
 Count   Count   Pages   Pages    MEMutl  MEMutl  MEMutl  MEMutl
 ------ ------ ------ ------     ------  ------  ------  ------
    53     50   2817    251      94.9%   93.6%   93.1%    2.1%

                                                        Paging Rates
  Page   System   Pages   Read   Pages   Write   Free   Modify
 Faults  Faults   Read    I/Os   Writen  I/Os    List    List
 ------  ------  ------ ------  ------  ------  ------  ------
  147.9    1.6     40.2    6.3    23.9     0.2     9.3    14.8
```

8.3 Isolating an I/O Limitation

The first step in isolating an I/O limitation is to determine if there is a hardware problem. A hardware problem can affect both buffered and direct I/O statistics. Type the following DCL command for a report of any hardware errors:

```
$ SHOW ERROR
```

An example of the report produced by this command is:

```
Device                Error Count
PAA0:                          7
GREEN$DUA2:                    1
GREEN$DUA15:                   2
YELLOW$MUA0:                   6
```

The report lists each device on your system that has recorded errors and the number of errors. A high error count for a device may indicate a hardware problem on that device. Correct any hardware problems and return to the diagnosis step in Chapter 7. If symptoms of an I/O limitation persist, you can proceed to isolate the cause of the limitation.

I/O on a VMS system is divided into two types: direct I/O and buffered I/O. Direct I/O involves the transfer of data directly to or from the process buffer; system buffers are not used. Buffered I/O involves a transfer to or from a system space buffer rather than a process buffer.

Direct I/O relates to disks, tapes, and high-speed links (for example, DR780). Buffered I/O relates to terminals, line printers, console disk drives, and communications devices (DMR11, DEQNA). A high buffered I/O rate can also be caused by misaligned direct I/O buffers and explicit buffered $QIOs.

8.3.1 Direct or Buffered I/O

A high direct I/O rate may be indicated in two ways on the tabular report: by a Direct I/O (system) rate that is greater than 30 times the CPU factor for any disk or device, or by a Rate(/s) that exceeds 15 for any disk. (See Table 7–2 for the CPU factor associated with your system.)

A high buffered I/O rate may be indicated by a Buffered I/O rate that is greater than 100 times the CPU factor. (See Table 7–2 for the CPU factor associated with your system.)

- High direct I/O rate—Proceed to Section 8.3.2.
- High buffered I/O rate—Proceed to Section 8.3.9.

8.3.2 Determining the Device for a Direct I/O Problem

Disks and tapes with direct I/O problems show long delay times for I/O completion.

Performance degradation implies one or both of the following situations:

- The device is not fast enough.
- The total demand on the device is so high that requests are blocked while others are being serviced.

The device I/O rates and disk statistics sections of the tabular report provide specific device and disk activity information. Note that if you do not explicitly specify device statistics when you collect and report the data (using the PERFORMANCE COLLECT and PERFORMANCE REPORT commands), they do not appear in the report. All disks are collected and reported by default unless they are explicitly negated in the command.

Look for devices that perform I/O with excessive rates. For example, in Figure 8–18 the Device I/O Rates (per second) section shows that disk mailbox XEA1 has an I/O rate of 30. This may be excessive for your system.

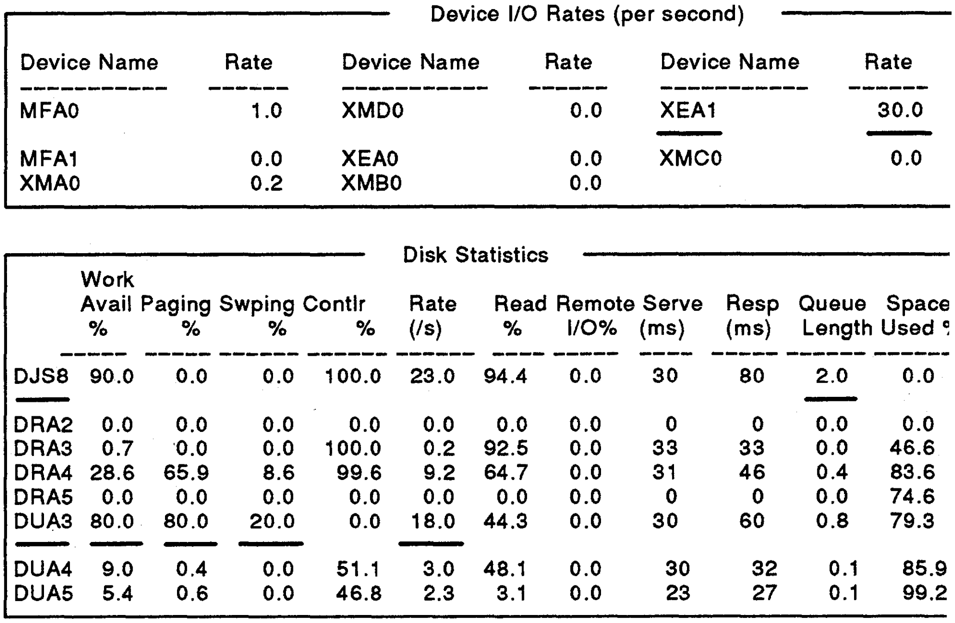
Look for disks with a queue of pending requests (Queue Length). In Figure 8–18, the Disk Statistics section indicates that disk DJS8 has a Queue Length of 2.

Look at Work Avail%, Paging%, Swping%, and Rate for each disk. Figure 8–18 shows that disk DUA3 has a Work Avail% of 80.0, a Paging % of 80.0, a Swping % of 20.0, and a Rate per second of 18.0, which may indicate a bottleneck on this disk.

An abnormally high device I/O rate suggests that the I/O demand for the device exceeds its capacity.

Examine your reports as described in the following sections to isolate the cause of an I/O limitation for a device.

Figure 8–18 Device I/O Rates and Disk Statistics Sections



```
                              Device I/O Rates (per second)

Device Name     Rate    Device Name     Rate    Device Name     Rate
-----------    ------   -----------    ------   -----------    ------
MFA0              1.0   XMD0              0.0   XEA1             30.0

MFA1              0.0   XEA0              0.0   XMC0              0.0
XMA0              0.2   XMB0              0.0

                              Disk Statistics
     Work
     Avail Paging Swping Contlr  Rate   Read Remote Serve  Resp  Queue  Space
      %      %      %      %     (/s)    %    I/O%  (ms)   (ms)  Length Used
     ----- ----- ------ ------ ------ ---- ----- ----  ----- ------ ------
DJS8  90.0   0.0    0.0  100.0   23.0 94.4   0.0   30     80    2.0    0.0

DRA2   0.0   0.0    0.0    0.0    0.0  0.0   0.0    0      0    0.0    0.0
DRA3   0.7   0.0    0.0  100.0    0.2 92.5   0.0   33     33    0.0   46.6
DRA4  28.6  65.9    8.6   99.6    9.2 64.7   0.0   31     46    0.4   83.6
DRA5   0.0   0.0    0.0    0.0    0.0  0.0   0.0    0      0    0.0   74.6
DUA3  80.0  80.0   20.0    0.0   18.0 44.3   0.0   30     60    0.8   79.3

DUA4   9.0   0.4    0.0   51.1    3.0 48.1   0.0   30     32    0.1   85.9
DUA5   5.4   0.6    0.0   46.8    2.3  3.1   0.0   23     27    0.1   99.2
```

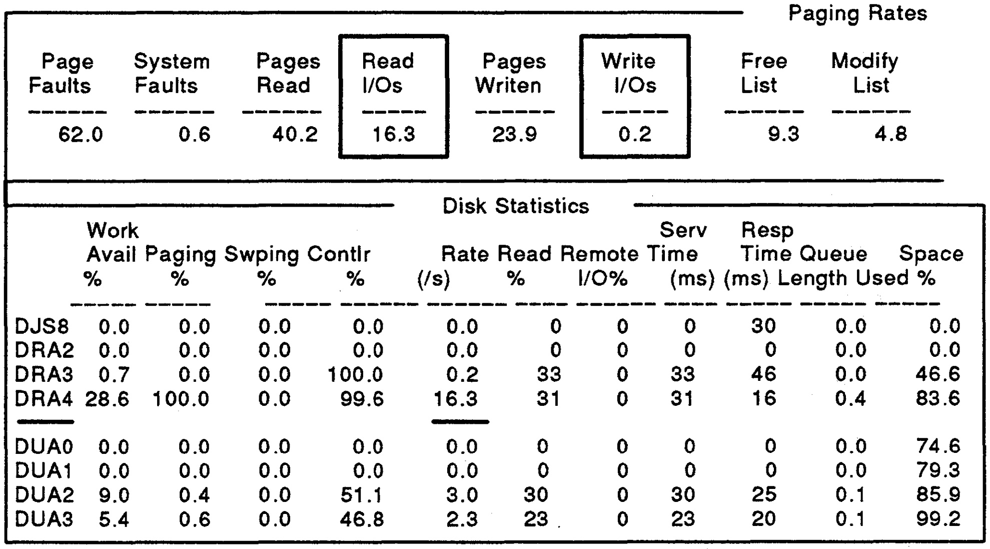
8.3.3 Excessive Paging or Swapping I/O

Look at the tabular report's Paging Rates and Disk Statistics sections.

First look at Read I/Os, which tell you how much disk activity is due to hard faulting. Then look at Write I/Os, which indicate how much modified page writing is occurring (part of swapper activity). For example, in Figure 8–19 the value of Read I/Os is 16.3 and the Write I/Os is 0.2.

If paging I/O or swapper activity seem high, as they do in Figure 8–19, the next step is to examine disk statistics and determine which disk has a high percentage of paging (Paging%) or swapping (Swping%) activity. Then look at Rate(/s) for the disk to see if this is a concern. In Figure 8–19 disk DRA4 has a Rate(/s) of 16.3, all of which is paging. This high rate per second may indicate that excessive paging is occurring on disk DRA4.

Figure 8–19 Paging Rates and Disk Statistics Sections of Tabular Report



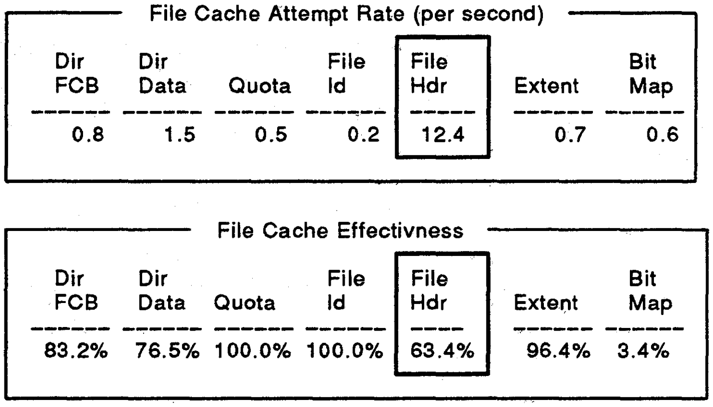
```
                                                                Paging Rates

  Page    System    Pages    Read      Pages     Write       Free    Modify
 Faults   Faults    Read     I/Os      Writen    I/Os        List     List
 ------   ------   ------    -----     ------    -----      ------   ------
   62.0      0.6     40.2     16.3       23.9     0.2          9.3      4.8

                              Disk Statistics
       Work                                                 Serv   Resp
       Avail Paging Swping Contlr     Rate Read Remote Time     Time Queue   Space
        %      %      %      %       (/s)    %    I/O%   (ms) (ms) Length Used %
      ----- -----   ----- ------ ------ ---- ------ ---- ----- ----- -----
DJS8   0.0    0.0    0.0     0.0    0.0     0     0     0    30    0.0     0.0
DRA2   0.0    0.0    0.0     0.0    0.0     0     0     0     0    0.0     0.0
DRA3   0.7    0.0    0.0   100.0    0.2    33     0    33    46    0.0    46.6
DRA4  28.6  100.0    0.0    99.6   16.3    31     0    31    16    0.4    83.6

DUA0   0.0    0.0    0.0     0.0    0.0     0     0     0     0    0.0    74.6
DUA1   0.0    0.0    0.0     0.0    0.0     0     0     0     0    0.0    79.3
DUA2   9.0    0.4    0.0    51.1    3.0    30     0    30    25    0.1    85.9
DUA3   5.4    0.6    0.0    46.8    2.3    23     0    23    20    0.1    99.2
```

8.3.4 Poor File System Caching

Too small a size for the file cache causes excessive ACP/XQP I/O operations; too large a size consumes physical memory. If the cache attempt rate is high but the cache effectiveness rate is low, then the cache is ineffective.

Look at the File Cache Attempt Rate and File Cache Effectiveness sections of the tabular report. In Figure 8–20, the File Hdr caching appears to be ineffective. The attempt rate of 12.4 means the cache is heavily used while the effectiveness rate is only 63.4%.

Figure 8–20 File Cache Attempt and Effectiveness Rate Section of Tabular Report



```
 ---------- File Cache Attempt Rate (per second) ----------
  Dir     Dir             File    File              Bit
  FCB     Data    Quota   Id      Hdr      Extent   Map
 ------  ------  ------  ------  -----    ------  ------
   0.8     1.5     0.5     0.2    12.4       0.7     0.6

 ----------------- File Cache Effectivness -----------------
  Dir     Dir             File    File              Bit
  FCB     Data   Quota    Id      Hdr      Extent   Map
 ------  ------  ------  ------  ----     ------  ------
 83.2%   76.5%  100.0%  100.0%  63.4%     96.4%    3.4%
```

If cache effectiveness is 70% or greater, caching activity is normal.

Note that a 100% cache hit rate may require a very large cache and is an inefficient use of memory. If you can obtain a 70% hit rate, do not keep increasing the cache size.

8.3.5 High Disk Activity

Look at the Disk Statistics section of a few days' tabular reports.

High disk activity may be present when any of the following statistics are high for one or more disks:

- Paging %
- Swping %
- Rate(/s)
- Serv Time (ms)

- Resp Time (ms)
- Queue Length

Figure 8–21 is an example of a Disk Statistics section of a tabular report showing high activity for disks DJS8 and DUA3 with rates of 30 and 35 per second, respectively.

If one or more of these statistics are high, the next step is to determine the files that are responsible for the disk activity. Use the PERFORMANCE DISPLAY=FILES command as described in Chapter 9 to identify these files.

Figure 8–21 Tabular Report Segment Indicating High Disk Activity

```
---------------------------------- Disk Statistics -------------------------
        Work
        Avail Paging Swping Contlr  Rate   Read Remote Serve  Resp Queue  Space
          %      %      %      %    (/s)     %   I/O%  (ms)   (ms) Length Used %
        ----- ------ ------ ------ ------  ---- ----- -----  ----- ----- ------
 DJS8   90.0    0.0    0.0  100.0   30.0    0.0   0.0    39     87   2.0   87.2
 ----                              ------
 DMA0    0.0    0.0    0.0    0.0    0.0    0.0   0.0     0      0   0.0    0.0
 DRA0    0.7    0.0    0.0  100.0    0.2    0.0   0.0    33     33   0.0   46.6
 DRB0   28.6   65.9    8.6   99.6    9.2    0.0   0.0    31     46   0.4   83.6
 DRB1    0.0    0.0    0.0    0.0    0.0    0.0   0.0     0      0   0.0   74.6
 DUA3   80.0   80.0   20.0    0.0   35.0    0.0   0.0    44     88   1.6   79.3
 ----                              ------
 DRD0    9.0    0.4    0.0   51.1    3.0    0.0   0.0    30     32   0.1   85.9
 DRD1    5.4    0.6    0.0   46.8    2.3    0.0   0.0    23     27   0.1   99.2
```

8.3.6 Disk Fragmentation

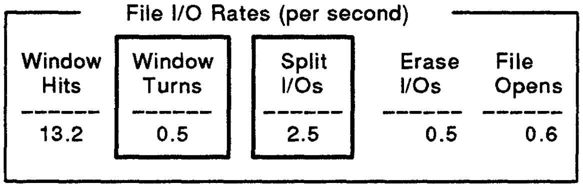
Accessing badly fragmented files increases the following activities:

- I/O operations
- Window turns
- Disk head movement

The presence of disk fragmentation may be indicated when Window Turns and Split I/Os on the tabular report are greater than 0.1. For example, in Figure 8–22 the Window Turns field has a value of 0.5 and Split I/Os field has a value of 2.5. This report may indicate the presence of disk fragmentation.

See Chapter 10 for a description of how to use the VAX SPM Disk Space Reporting facility to further investigate disk fragmentation.

Figure 8–22 Tabular Report Segment Indicating Disk Fragmentation



```
            File I/O Rates (per second)
 Window   Window    Split     Erase   File
  Hits    Turns     I/Os      I/Os    Opens
 ------   ------   ------    ------  ------
  13.2     0.5      2.5        0.5     0.6
```

8.3.7 High-Water Marking

High-water marking is a file security measure that is enabled by VMS by default. If it is not necessary on your system, disabling it may improve disk performance. Look at the File I/O Rates section of the tabular report. Any Erase I/Os may indicate the presence of high-water marking on your system. In Figure 8–23 the value of the Erase I/Os field is 0.5, which indicates high-water marking.

Once you have determined that high-water marking is present on your system, type the following command to see which disks have high-water marking enabled:

```
$ SHOW DEVICE/FULL
```

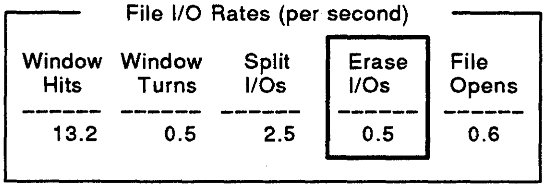
The Volume status line lists high-water marking if it is present on the disk. Figure 8–24 is an example of output of the DCL command SHOW DEVICE, showing high-water marking enabled on Disk DBA0.

Type the following command to disable high-water marking:

```
$ SET VOLUME device-spec[:][,...]/NOHIGHWATER_MARKING
```

For this command to take effect, you must dismount and remount the disk for each CPU in the VAXcluster system.

Figure 8–23 File I/O Rates Section of Tabular Report



```
            File I/O Rates (per second)
 Window  Window   Split    Erase    File
  Hits    Turns   I/Os     I/Os     Opens
 ------  ------  ------   ------   ------
  13.2    0.5     2.5      0.5      0.6
```

Figure 8–24 Device Report

```
Disk DBA0:, device type RP06, is online, mounted, file-oriented device,
    shareable, error logging is enabled.

Error count                    2    Operations completed            820534
Owner process                 ""    Owner UIC                     [SYSTEM]
Owner process ID        00000000    Dev Prot     S:RWED,O:RWED,G:RWED,W:RWED
Reference count              165    Default buffer size                512
Total blocks              340670    Sectors per track                   22
Total cylinders              815    Tracks per cylinder                 19

Volume label       "VAX5SYSDPAK"    Relative volume number               0
Cluster size                   1    Transaction count                  183
Free blocks                40473    Maximum files allowed            85167
Extend quantity                5    Mount count                          1
Mount status              System    Cache name              "_DBA0:XQPCACHE"
Extent cache size             64    Maximum blocks in extent cache    4047
File ID cache size            64    Blocks currently in extent cache    84
Quota cache size               0    Maximum buffers in FCP cache       506

Volume status:  subject to mount verification, file high-water marking,
write-through caching enabled.
```

8.3.8 Explicit Direct QIOs

Look at the Process Metrics report and examine Direct I/O rates for each process. Figure 8–25 is an example of a Process Metrics report.

When you find a process that is executing a lot of direct I/Os, determine whether the process is executing a program that employs explicit Direct QIOs (other than RMS). If this process is using a disk that has an excessive I/O rate, the cause of the excessive I/O rate may be this process. Process F_USER in Figure 8–25 is executing a lot of direct I/Os.

A program that issues a lot of direct QIOs may not be designed efficiently. Redesign of the program may include caching to improve the efficiency of QIOs.

8.3.9 Determining the Device and Process — Buffered I/Os

Excessive buffered I/Os show up in two places:

- Device I/O rates (terminal, line printer, mailbox, etc.)
- Process metrics

From Device I/O Rates, determine which device is responsible. For example, in Figure 8–26 device PURPLE$XEA1 has an I/O rate of 32.2, which may indicate an excessive I/O rate.

Figure 8–25 Identify the Process

```
----------------------------------------- Process Metrics ------------------------
                             //
                             \\                                    Direct I/O Buffrd I/O
Process ID Process Name      //  Process Pri   CPUtime(min)      /sec        /sec
-------- ------------        \\  ----------- ------------      -------  ---------
20200302  V72:A_USER         //       8             0.004        0.243      0.303
20200304  V73:B_USER         \\       8             0.000        0.000      0.000
20200220  V54:C_USER         //       5             0.160        3.093      1.395
202000A1  SPM_D_USER         \\      24             0.000        0.000      0.000
20200329  SPM_E_USER         //      24             0.002        0.303      0.061
202002AD  V59:F_USER         \\       8             1.124       29.243      2.062
                             //
202002B0  V60:G_USER         \\       8             0.000        0.000      0.000
202002BC  V65:H_USER         //       8             0.000        0.000      0.000
```

Figure 8–26 Device Statistics Section of Tabular Report

```
---------------------- Device I/O Rates (per second) -----------------------

Device Name     Rate    Device Name     Rate    Device Name      Rate
-----------     -----   -----------     ------  -----------      ----
PURPLE$LPA0      0.0    PURPLE$MTA0      0.0    PURPLE$OPA0       0.0
PURPLE$TTA6      0.0    PURPLE$XEA1     32.2
```

Figure 8–27 Tabular Report Segment Showing a Device with High I/O Rate

```
---------------------------  \\  ---------  Process Metrics  ------------------
                             //
                             \\                                 Direct I/O Buffrd I/O
Process ID Process Name      //    Process Pri CPUtime(min)       /sec        /sec
---------- ------------      \\  ----------- ------------  ----------  -------
20200302  V72:A_USER         //        8          0.006          0.023      0.105
20200304  V73:B_USER         \\        5          0.936          0.295     32.200
                             //
20200220 V54:C_USER          \\        5          0.240          1.145      1.498
202000A1 SPM_CAPACITY        //       24          0.000          0.000      0.000
20200329 SPM_TUNE            \\       24          0.003          0.421      0.121
202002AD V59:D_USER          //        8          0.004          0.135      1.341
202002B0 V60:E_USER          \\        8          0.000          0.000      0.000
202002BC V65:F_USER          //        8          0.000          0.000      0.000
```

Examine the Process Metrics report to determine which process is causing the high I/O rate, then investigate that process. For example, in Figure 8–27 process B_USER has an I/O rate of 32.2 per second.

8.4 Isolating a CPU Limitation

Look at the tabular report. A CPU limitation may be indicated by any of the following circumstances:

- The CPU queue is greater than 1.0.
- Idle Time is less than 15%.
- System CPU time is greater than Task and Task is greater than 25%.

The following sections describe how to examine your reports to isolate the cause of a CPU limitation.

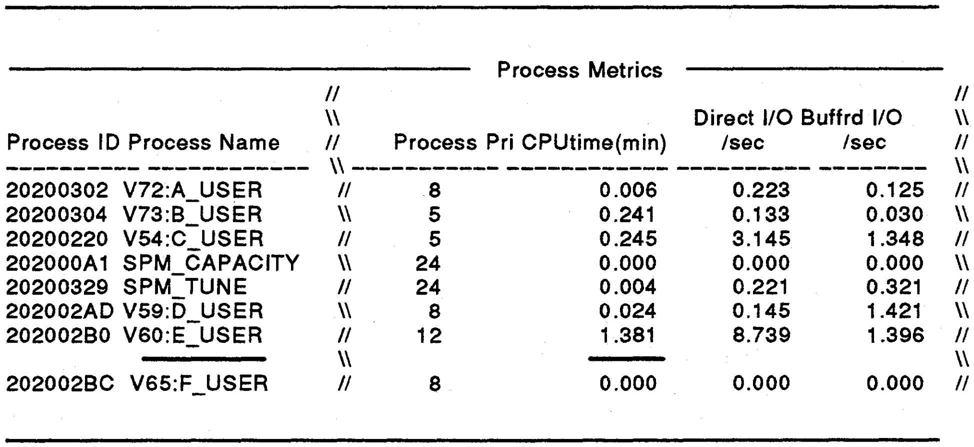
8.4.1 Processes Blocking Other Processes

A high-priority process may be responsible for blocking lower priority processes. Two situations where blocking may be observed are:

- The high-priority process may be running an inefficient program.
- A server, or process with which other processes must communicate, might be a bottleneck.

Look at the process metrics to see if any process or group of processes are monopolizing the CPU. For example, in Figure 8–28 process E_USER is using 1.381 minutes of CPU time and has a priority of 12. In this example, process E_USER may be blocking processes A_USER, B_USER, C_USER, and D_USER. Changing the priority of process E_USER may alleviate the bottleneck.

Figure 8–28 Process Metrics Report Section



```
--------------------------------------- Process Metrics ----------------------------
                               //                                                  //
                               \\                          Direct I/O Buffrd I/O   \\
Process ID Process Name        //   Process Pri CPUtime(min)   /sec      /sec      //
---------- ------------        \\ ----------- ------------ ---------- --------     \\
20200302  V72:A_USER           //      8           0.006      0.223     0.125      //
20200304  V73:B_USER           \\      5           0.241      0.133     0.030      \\
20200220  V54:C_USER           //      5           0.245      3.145     1.348      //
202000A1  SPM_CAPACITY         \\     24           0.000      0.000     0.000      \\
20200329  SPM_TUNE             //     24           0.004      0.221     0.321      //
202002AD  V59:D_USER           \\      8           0.024      0.145     1.421      \\
202002B0  V60:E_USER           //     12           1.381      8.739     1.396      //
             ---------         \\                  ------                          \\
202002BC  V65:F_USER           //      8           0.000      0.000     0.000      //
```

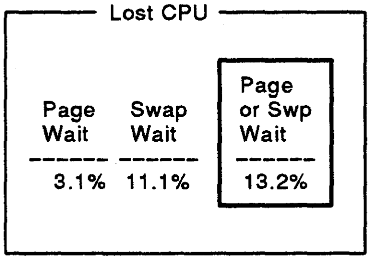
8.4.2 Lost CPU Time

Is CPU time being "lost" because the CPU has to wait for disk transfers for page or swap I/O? Look at Page or Swp Wait in the Lost CPU section of the tabular report.

A value for Page or Swp Wait of 1% would not be significant, whereas a value of 10% would be cause for concern. Remember, a memory or an I/O problem can manifest itself as a CPU problem. In Figure 8–29, the Page or Swp Wait field value of 13.2% may indicate a problem.

The VAX SPM PC Analysis facility (Chapter 11) provides a detailed view of the program counter. Use the VAX SPM PC Analysis facility to further isolate the cause of lost CPU time.

Figure 8–29 "Lost" Time

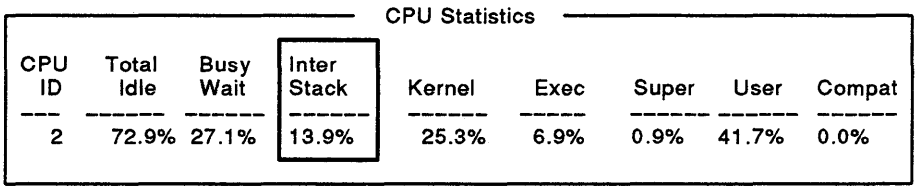


```
------- Lost CPU -------
                   Page
 Page   Swap       or Swp
 Wait   Wait       Wait
------ ------     ------
  3.1%  11.1%      13.2%
```

8.4.3 Excessive Interrupt Stack Activity

Look at the CPU Statistics section of the tabular report. A value of 10% for Inter Stack is moderate, a value of 20% or more on processors other than VAX-11/780 systems is excessive. For example, in Figure 8–30 the value of 13.9 for the Inter Stack field does not indicate a problem.

Figure 8–30 Interrupt Stack Activity



```
                              CPU Statistics
CPU   Total   Busy    Inter
 ID   Idle    Wait    Stack     Kernel    Exec    Super   User    Compat
---  ------  ------  -------   -------   -----   ------  ------   ------
 2   72.9%   27.1%   13.9%      25.3%    6.9%    0.9%    41.7%    0.0%
```

Blocking occurs when one high-priority process prevents others from executing. Blocking may not be due to contention of processes with other processes at higher priorities, but due to too much activity on the interrupt stack. If the rate of interrupts is excessive, it may prevent processes from using the CPU.

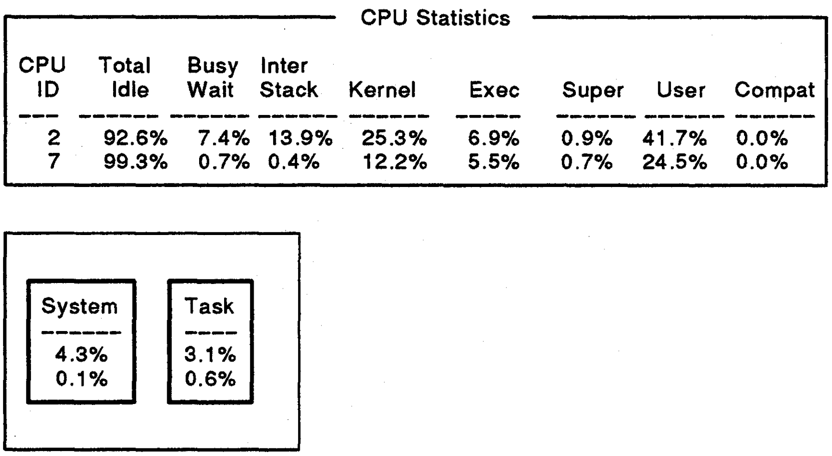
The System Module Usage portion of a System PC report filters Interrupt Stack activity against the system module list when /IDENTIFICATION is omitted. This report can help to identify the causes of excessive interrupts. Chapter 11 describes the System PC Analysis facililty and how to interpret System PC reports.

Examine the PC Analysis reports to learn which devices cause excessive interrupts, and how the rate might be reduced. If terminals are the cause of excessive interrupt stack activity, system performance might benefit by using DMF32s instead of DZ11s or DZ32s, so that terminal I/O is transferred in a burst.

8.4.4 Excessive System CPU Usage

Look at the tabular report fields System and Task (under CPU Statistics). The System value is the CPU time the system uses to keep itself running; it can be regarded as overhead. The Task value is the amount of CPU time the system uses to perform work. In most cases, the system should be devoting more resources to getting the job done (task) than in overhead (system). When the system is not busy, resources devoted to system can exceed task and not present a problem. If System is greater than Task, and Task is greater than 25% of total CPU time, however, there may be excessive CPU overhead usage. In Figure 8–31, System on one processor is 4.3 and Task is 3.1. Although the system is using a greater percentage of CPU than the task, since task is below 25%, a problem is not indicated.

Figure 8–31 Tabular Report Segment for Isolating Excessive System CPU Use



```
                         CPU Statistics

CPU   Total   Busy  Inter
 ID    Idle   Wait  Stack   Kernel     Exec    Super   User  Compat
---  ------ ------ ------  ------    -----   ------ ------ ------
  2   92.6%   7.4%  13.9%   25.3%     6.9%    0.9%  41.7%   0.0%
  7   99.3%   0.7%   0.4%   12.2%     5.5%    0.7%  24.5%   0.0%


 System     Task
 ------     ----
  4.3%      3.1%
  0.1%      0.6%
```

If excessive system CPU usage is the problem, the next step is to determine who is using the CPU, at what access modes, and for what reason. The VAX SPM System PC Analysis utility provides this information. Chapter 11 describes the VAX SPM System PC Analysis facililty and how to use it to further isolate the causes of excessive CPU usage.

9 Disk Activity Analysis

This chapter provides the following information about VAX SPM File Activity display:

- Introduction to the File Activity display
- Instructions for invoking the File Activity display
- Description of File Activity display interactive commands

9.1 Introduction

When an I/O bottleneck is due to high activity on a disk, the information from the File Activity display can help you to improve system performance. The File Activity display identifies the files on a disk that are most frequently accessed. In addition, it displays the percentage of read operations, I/O rate, and usage for these files. Depending on the workload, these files may be moved to less busy disks to reduce the I/O on the excessively active disk. Active files may also be marked for allocation of RMS GLOBAL buffers.

9.2 Invoking the File Activity Display

The privileges required to use the PERFORMANCE DISPLAY=FILES command are CMKRNL, TMPMBX, and SYSPRV.

The command is of the following type:

```
$ PERFORMANCE DISPLAY=FILES/qualifiers[,...]
```

There are no parameters for the command and there are three qualifiers as shown in Table 9–1.

Table 9–1 File Activity Command Qualifiers

Command Qualifier	Defaults
/DISK=([disk,...])	/DISK=SYS$SYSDEVICE
/INTERVAL=[seconds]	/INTERVAL=10
/VERSION	Not present in command line

Use the /DISK qualifier to specify the names of the disks whose activity you wish to monitor. Select disks that show chronically high I/O rates or response times as described in Section 8.3.5.

Use the /INTERVAL qualifier to specify the interval at which file activity information is displayed. If no interval is specified, an interval of 10 is assumed.

File activity data is collected at a rate computed as the display rate divided by 10. It is always greater than or equal to 1 second, and less than or equal to 5 seconds. Due to the small collection interval, collecting data for a number of disks over a long period may affect overall system performance.

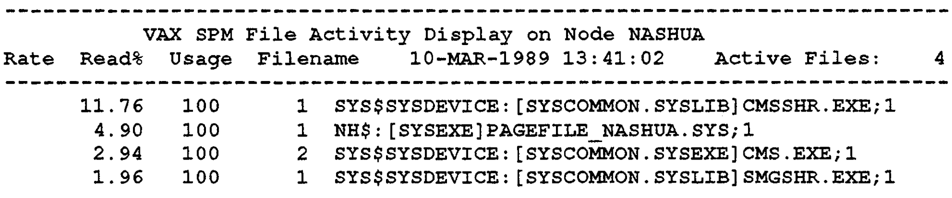
The /VERSION qualifier displays the version number of this utility.

The following command displays active files for disks DUA0 and SYS$DISK: at the default rate of 10 seconds.

```
$ PERFORMANCE DISPLAY=FILES/DISK=(DUA0,SYS$DISK)
```

Figure 9–1 is an example of the File Activity display. Files are listed in descending order according to their read/write operation rate.

Figure 9–1 File Activity Display



```
--------------------------------------------------------------------------
           VAX SPM File Activity Display on Node NASHUA
Rate  Read%  Usage  Filename    10-MAR-1989 13:41:02    Active Files:    4
--------------------------------------------------------------------------
      11.76   100      1  SYS$SYSDEVICE:[SYSCOMMON.SYSLIB]CMSSHR.EXE;1
       4.90   100      1  NH$:[SYSEXE]PAGEFILE_NASHUA.SYS;1
       2.94   100      2  SYS$SYSDEVICE:[SYSCOMMON.SYSEXE]CMS.EXE;1
       1.96   100      1  SYS$SYSDEVICE:[SYSCOMMON.SYSLIB]SMGSHR.EXE;1
```

Table 9–2 describes fields in the File Activity display.

Table 9–2 Fields in the File Activity Display

Field	Description
Rate	The number of read and write operations per second for the file
Read %	The percentage of Read I/Os attributed to file reads
Usage	The number of concurrent references to the file
Name	The full name of the active file, including its device (which may be a user-defined logical)
Active Files	The total number of active files on the specified disk(s) with a rate greater than 0

If a disk dismounts while DISPLAY=FILES is running, a message displays and collection discontinues for that disk. If the disk is remounted, it must be respecified before its performance statistics are collected and displayed.

9.3 File Activity Interactive Commands

Table 9–3 describes the File Activity display interactive subcommands.

Table 9–3 File Activity Interactive Subcommands

Subcommand	Meaning
EXIT or CTRL/Z	Exits from the display
HELP	Displays HELP information for field descriptions and subcommands
SET DISK	Defines the disks for which statistics are displayed
SET INTERVAL	Defines the rate at which disk statistics are displayed
SHOW DISK	Shows the names of disks for which statistics are displayed
SHOW INTERVAL	Shows the value of the current display interval
SHOW ALL	Shows the value of the current display interval and the names of disks for which statistics are displayed
SPAWN	Spawns a subprocess
WRITE [file-spec]	Copies the screen contents to a file. If no file is specified, the default of SPM$DISPLAY_FILES.RPT is used

The *VAX SPM Reference Manual* describes each subcommand.

10 Disk Space Analysis

This chapter provides the following information about the VAX SPM Disk Space Analysis utility:

- Introduction to the Disk Space Analysis utility
- Instructions for using the Disk Space Analysis utility
- Instructions for determining if there is fragmentation on a disk
- Instructions for determining if disk use corresponds to its initialization

This chapter provides general descriptions and examples, and the *VAX SPM Reference Manual* provides detailed descriptions of command syntax and reports.

10.1 Introduction

When an I/O bottleneck is due to disk fragmentation, the information from the Disk Space Analysis utility can help you to improve system performance. Use the Disk Space report to determine if there is disk fragmentation, the disks on which it occurs, and the files on the disk that are most affected. Once you have determined the disks that are fragmented, you can compress them using the VMS BACKUP utility.

The Disk Space report can also help you determine if you are getting maximum utilization from your disks by showing any inconsistencies between how a volume is initialized and mounted and how it is used. For example, you can see if the extension size is too small or too large in relation to the size of most files on the disk. If the extension size is inappropriate, adjusting it may optimize disk utilization.

10.2 Using the Disk Space Analysis Utility

Use the PERFORMANCE REPORT=DISK_SPACE command to obtain information about disk volumes. The following are prerequisites:

- The volume must be in FILES-11 ODS-2 format.
- The volume must be mounted without the /FOREIGN qualifier.
- You must have READ access to the files [0,0]BITMAP.SYS and [0,0]INDEXF.SYS on the volume to be analyzed. Generally, this is satisfied by the SYSPRV privilege.

The report command is of the following type:

```
$ PERFORMANCE REPORT=DISK_SPACE/qualifier device-name,...
```

This command accepts one or more parameters and a qualifier (explained below). The parameters are the names of disk devices on which to report. Physical or logical device names may be specified.

Select disks that show the most activity in the Disk Statistics section of the tabular report, or disks that are more than 70% full. Before giving the REPORT command, you can display a list of disk devices on your system using the DCL command SHOW DEVICES.

During the analysis, the volume is locked to prevent storage allocation. The files BITMAP.SYS and INDEXF.SYS are read and the volume is unlocked. The command does not write to the volume being analyzed unless the report file you specify resides on that volume.

By default, the disk information is written to a report file called DISKSPACE.RPT. Using the /OUTPUT qualifier, you can specify the output report name.

The following command writes a disk volume report for devices DBA0: and DBA1: to the file DISKSPACE.RPT in the current directory:

```
$ PERFORMANCE REPORT=DISK_SPACE  DBA0,DBA1
```

The following command writes a disk volume information report for device HSC000$DUA0 to the file DUA0.RPT:

```
$ PERFORMANCE REPORT=DISK_SPACE/OUTPUT=DUA0.RPT HSC000$DUA0
```

The Disk Space report contains four sections: Detailed Volume Analysis, Summary of Free Storage, Summary of Allocated Space, and Files with Extension Headers. Figure 10–1 through Figure 10–4 are examples of these sections.

10.3 Investigating Disk Fragmentation

There are three ways to read the Disk Space report to evaluate disk fragmentation. One way is to compare the size of most of the current files with the size of most available space to determine if fragmentation will occur as new files are created on the disk. Another way is to look at the number of files on a disk and the size and number of extents per file to determine if files on the disk are fragmented. The third way is to look at the number of files with extension headers to determine if the disk is fragmented.

To determine if fragmentation will occur as new files are created, examine the Free Storage Utilization section of the Disk Space report. Look at the free storage extent size closest to 80%. Now examine the Allocated Space Utilization section of the Disk Space report. Look at the allocated extent size closest to 80%. If the size of most space allocated (files created) is smaller than most free storage size (space available), little or no fragmentation is likely to occur on that disk. If the size of most files created is larger than the free extent size, fragmentation may occur as new files are created.

In Figure 10–1, in the 82.5 percentile, free storage extent sizes range from 900 to 1500 blocks. In Figure 10–2, in the 84.6 percentile, allocated extent sizes range from 60 to 90 blocks. Therefore, the size of free storage is more than enough to accommodate the new files as they are created, and disk fragmentation is not likely to occur if future disk usage follows current usage.

Figure 10–1 Free Storage Utilization—Report Example

```
1  9-JUN-1988 10:49:47.66                              VAX SPM V3.3     Page   1
 *****  Summary of FREE STORAGE for    _IOTECH$DUA1: *****

  Free Storage  Extent Sizes      No. Extents     Cum % Space
  -------------------------       -----------     -----------

  >=         3,  <         6          201             1.3
  >=         6,  <         9           88             2.4
  >=         9,  <        15           91             4.5
  >=        15,  <        30          110             9.2
  >=        30,  <        60           80            16.0
  >=        60,  <        90           42            22.5
  >=        90,  <       150           22            27.8
  >=       150,  <       300           36            43.6
  >=       300,  <       600           16            57.8
  >=       600,  <       900           10            74.4
  >=       900,  <      1500            3            82.5
  >=      1500,  <      3000            2            91.0
  >=      3000,  <      6000            1           100.0
  >=      6000,  <      9000            0           100.0
  >=      9000,  <     15000            0           100.0
  >=     15000,  <     30000            0           100.0
  >=     30000,  <     60000            0           100.0
  >=     60000,  <     90000            0           100.0
  >=     90000,  <    150000            0           100.0
  >=    150000                          0           100.0

 Total free blocks  =       46284.
 No. of extents     =         702.
 Mean blocks/extent =          66.
 Smallest extent    =           3.
 Largest extent     =        4158.
```

To evaluate file fragmentation, examine the Allocated Space section of the Disk Space report. Look at the number of extents allocated, the total number of files, and the mean number of extents per file. A value of 1 for mean number of extents per file indicates that most files have only one extent; therefore, they are not fragmented. A disk with a small number of files and a large number of extents per file indicates that the files are fragmented. In Figure 10–2, the mean number of extents per file is 2 and there are a large number (10052) of small files, indicating that these files may be fragmented.

To further determine if a disk is fragmented, look at the number of extension headers in the Allocated Space section of the Disk Space report. When the number of extension headers is nonzero, there are one or more seriously fragmented files. A single header holds about 80 extent pointers; therefore, a file that needs an extension header is severely fragmented. In Figure 10–2 there are 26 extension headers, which indicates this disk is heavily fragmented.

Figure 10–2 Allocated Space Utilization—Report Example

```
1 9-JUN-1988 10:49:47.66                        VAX SPM V3.3    Page   2
*****  Summary of ALLOCATED SPACE for   _IOTECH$DUA1: *****

  Space Allocated per Header     No. Headers     Cum % Headers
  --------------------------     -----------     -------------

  >=          3,  <         6          2979            29.7
  >=          6,  <         9          1162            41.2
  >=          9,  <        15          1116            52.3
  >=         15,  <        30          1423            66.5
  >=         30,  <        60          1296            79.4
  >=         60,  <        90           522            84.6
  >=         90,  <       150           742            92.0
  >=        150,  <       300           448            96.4
  >=        300,  <       600           202            98.5
  >=        600,  <       900            34            98.8
  >=        900,  <      1500            37            99.2
  >=       1500,  <      3000            44            99.6
  >=       3000,  <      6000            18            99.8
  >=       6000,  <      9000            12            99.9
  >=       9000,  <     15000             7           100.0
  >=      15000,  <     30000             3           100.0
  >=      30000,  <     60000             0           100.0
  >=      60000,  <     90000             0           100.0
  >=      90000,  <    150000             0           100.0
  >=     150000                           0           100.0

 Minimum allocated extent =        3.
 Maximum allocated extent =    19191.
 Total allocated blocks   =   844731 ( 94.8% of volume).
 Total used      blocks   =   777058 ( 92.0% of allocated).
 No. extents allocated    =    18333.
 Mean alloc blocks/extent =       46.
 Total no. of files       =    10052.
 Mean alloc blocks/file   =       84.
 Mean no. extents/file    =        2.
 No. extension headers    =       26.
 No. multi-volume files   =        0.
 No. directories          =      435.
```

The Disk Space report lists any files with extension headers. Figure 10–3 is an example.

Figure 10–3 Files with Extension Headers—Report Example

```
1 9-JUN-1988 10:49:47.66                            VAX SPM V3.3     Page   3
*****  Files with extension headers for _IOTECH$DUA1: *****
  File name                                                    Ext. headers
  ------------------------------------------------------  ------------
  [A_USER]OBSAA_19880531_1200.EVM;1                               2
  [B_USER]OBSAB_19880531_1200.EVM;1                               5
  [C_USER]H.DAT;1                                                 1
  [D_USER]K.DAT;1                                                 9
  [E_USER]SPTESTSPMDAT.25;1                                       1
  [A_USER]OBSAC_19880531_1200.EVM;1                               4
  [A_USER]OBSAD_19880531_1200.EVM;1                               2
  [X]XLIB_REFERENCE_FT1.LN03;1                                    2
```

10.4 Determining if Disk Use Corresponds to Initialization

To determine if disk usage corresponds to the way it is initialized and mounted, examine the Detailed Volume Analysis section. Note the actual values of items affected by initializing and mounting, and compare them to the values they were assigned when the disk was initialized and mounted.

In the Detailed Volume Analysis section, items preceded by an "I" are controlled by disk initialization. Items preceded by an "M" are controlled by disk mounting, and items preceded by an "S" are controlled by system parameters. In Figure 10–4, the item Default file extension is preceded by both I and M. Therefore, disk initialization and mounting control this item.

Compare the value of the default file extension to the value of mean allocated blocks per extent in the Allocated Space section. If the value of the default file extension is significantly larger than the mean allocated blocks, then files are allocated more blocks than they actually need. Reinitializing the disk and reducing the default file extension size may improve disk utilization.

Figure 10–4 Detailed Volume Analysis Section of Disk Space Report

```
1 9-JUN-1988 10:49:47.66                         VAX SPM V3.3     Page    1

*****  Detailed volume analysis for     _IOTECH$DUA1:  *****

 Items preceded by 'I','M' or 'S' are controlled by Initialize,
 Mount or Sysgen.

 (I  ) Volume name  is 'SPMDEV      '.
 (I  ) Serial number is          0.
 (I  ) Creation date was  8-JAN-1988 14:54:05.19.
 (I  ) Volume owner is 'SYSTEM      '.
 (IM ) Owner uic is [00001,000004].
 (I  ) Format type  is 'DECFILE11B  '.
 (IM ) Volume protection is [RWED,RWED,RWED,RWED].
 (IMS) Default data checking is NOREAD-CHECK, NOWRITE-CHECK.
 (I  ) Structure level is   2, version   1.
 (I  ) Allocation cluster size is     3 blocks.
 (I  ) Index file bitmap is located at LBN     445614.
 (IM ) Default file extension is     5 blocks.
 (IM ) Default window size is    7 retrieval pointers.
 (I  ) Maximum number of files allowed is     111384.
 (IMS) Default number of cached directories is    3.

       Volume size is     891072 blocks with
                              51 blocks/track,
                              14 tracks/cylinder,
                            1248 cylinders/volume.
```

11 System Program Counter Analysis

This chapter provides instructions for performing the following tasks using VAX SPM System Program Counter (PC) Analysis:

- Collecting system-wide PC data using the PERFORMANCE COLLECT=SYSTEM_PC command
- Reporting on the collected data using the PERFORMANCE REPORT=SYSTEM_PC command
- Using System PC data to investigate how system CPU time is spent
- Using System PC data to isolate the cause of High Interrupt Stack activity

11.1 Introduction

System-wide PC (program counter) statistics can be used as a detailed CPU measurement tool. Whenever a performance investigation indicates that CPU usage is high or that the CPU may be a bottleneck, system-wide PC statistics can pinpoint exactly where (PC) and how [IPL (interrupt priority level) or access mode] the CPU is being used. System-wide PC statistics can be used to answer the following questions:

- Why is kernel mode time so high?
  - Is the time attributable to any particular process?
  - If so, is the time being spent calling system services or is a lot of code executed in kernel mode?
  - What amount of kernel time is accounted for in driver code or FDT routines? (FDT routines perform the device-dependent processing of an I/O request.)
  - What amount of CPU time is spent in each driver and in RMS routines?
- A real-time system is unable to meet its processing requirements but CPU utilization is low. The following questions may come to mind:
  - Are real-time interrupts being blocked by code executing at elevated IPL?
  - Are time-critical parts of user-mode code being stretched by elevated IPL code executing in response to device interrupts, fork processing, I/O post-processing, or AST delivery?

In Chapter 8, you used the tabular and Process Metrics reports to isolate resource limitations. If you encounter any of the following symptoms, you may want to use the PC Analysis facility to further investigate CPU usage.

- Lost CPU time indicated by a value greater than 10% for the Pg+Swp Wait field
- High Interrupt Stack activity indicated by a value of the Inter Stack field in the tabular report greater than 20%
- Excessive CPU usage indicated by a value of less than 15% Total Idle Time on the tabular report

The following sections tell you how to collect and report system-wide PC data, and provide examples of evaluating System PC reports to investigate the symptoms. The *VAX SPM Reference Manual* provides detailed descriptions of the PC Analysis commands and reports.

11.2 Using the System PC Analysis Facility

An overview of the system-wide PC data collection and reporting procedure is shown in Figure 11–1. The first step is to collect System PC data. This data is written to a log file. The next step is to run a report of System PC data.

To begin collecting system-wide PC data, you must give a command of the following type:

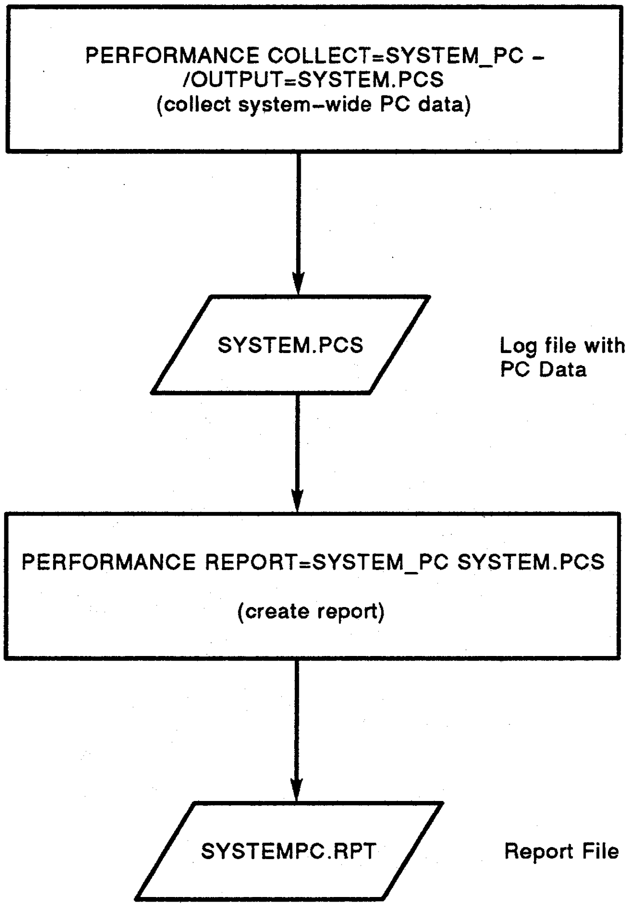
```
$ PERFORMANCE COLLECT=SYSTEM_PC/qualifiers
```

To use this command, you need either the SETPRV account privilege or all the following privileges:

```
ALTPRI    CMKRNL    PSWAPM    WORLD
```

A PC data collection image runs as part of your process and proceeds to collect PC data as specified by the command qualifiers. Since data is collected by your own process, you are not free to give other commands at your terminal while PC data is being collected. For this reason, it may be preferable to use a batch command file or to spawn another process for this command.

Figure 11–1 Collecting/Reporting System-Wide PC Data


PERFORMANCE COLLECT=SYSTEM_PC –
/OUTPUT=SYSTEM.PCS
(collect system–wide PC data)
SYSTEM.PCS
Log file with
PC Data
PERFORMANCE REPORT=SYSTEM_PC SYSTEM.PCS
(create report)
SYSTEMPC.RPT
Report File

The VAX SPM System PC Analysis facility collects a significant amount of data. Therefore, log files can get rather large. It is recommended that collections be limited to under 15 minutes for most investigations. If you need to collect System PC Analysis data for a longer period, refer to the *VAX SPM Reference Manual* to learn how to estimate the size of a System PC log file and ensure that you have adequate disk space.

As another example, the following command starts collecting system-wide PC statistics for all active processes at the default collection interval of 1 tick (10 milliseconds) and writes data to the PC log file SYSTEM.PCS in directory [ACCOUNT] on disk device DISK$. The collection terminates after the default period of 15 minutes.

```
$ PERFORMANCE COLLECT=SYSTEM_PC-/OUTPUT=DISK$:[ACCOUNT]SYSTEM.PCS
```

There can be only one process on your system at any given time collecting system-wide PC data. If such a process already exists when you give the above command, no new PC data collection image is started, and you receive the message:

```
%SPM-F-PCSAMPACT, PC sampling is currently active
```

You can stop your system-wide PC collection process at any time by typing CTRL/Y, or with the DCL STOP command issued from another process.

```
$ PERFORMANCE COLLECT=SYSTEM_PC/OUTPUT=PC.DAT
 Interrupt                              ;CTRL/Y was typed here
$
```

When you type a CTRL/Y, the PC log file is closed. To resume the collection process, type the DCL CONTINUE command before typing a command that runs a different image. If the CONTINUE command is used, no new version of the log file is created.

You can inquire on the system-wide PC data collection process by typing CTRL/T at the terminal from which it is running, or by using the DCL command SHOW PROCESS from another terminal.

11.2.1 Collecting PC Statistics for a Single Process

To collect PC statistics for a single process, use the /IDENTIFICATION qualifier with the PERFORMANCE COLLECT=SYSTEM_PC command and specify the PID (Process ID) for the selected process. You can determine the PID by using the DCL commands SHOW SYSTEM or SHOW USERS. This results in a much smaller PC log file and somewhat different reports.

In the following command, PC statistics are collected for the process whose ID is 00000054:

```
$ PERFORMANCE COLLECT=SYSTEM_PC/IDENTIFICATION=54/OUTPUT=PC54LOG.PCS
```

11.2.2 Generating Reports from a System-Wide PC Log File

To generate reports from a system-wide PC log file, you must give a command of the following type:

```
$ PERFORMANCE REPORT=SYSTEM_PC/qualifiers pclogfile
```

This command accepts one parameter, the name of the PC log file, and certain qualifiers as explained below. No special privileges are needed to use this command. The /OUTPUT qualifier can be used to give the file specification for the report; if this is omitted, the report is generated as SYSTEMPC.RPT in the user default directory.

The command below generates a report, SYSTEMPC.RPT, from the log file SYSTEM.PCS. The report contains statistics for a PC location, processor access mode, IPL level, interrupt stack activity, and processes active during the collection period.

```
$ PERFORMANCE REPORT=SYSTEM_PC  SYSTEM.PCS
```

The following command causes the same report to be generated as PC.RPT:

```
$ PERFORMANCE REPORT=SYSTEM_PC/OUTPUT=PC.RPT PCL001.DAT
```

11.2.3 Investigating System CPU Time

If there is excessive system CPU time on your system, use the video displays to monitor system performance. When a period of excessive system time occurs, run System PC collections.

Generate and examine the report to determine who is using the CPU, at what access modes, and for what reason. Figure 11–2 is an example of a System PC Analysis report segment. This report shows processor usage by process, the number of PC samples, and the total CPU and system time for each process. In this figure, most of the processes are responsible for high system CPU time as indicated by the numbers in the System Time % column that range from 72% to 100%. These processes also exhibit high kernel mode time as indicated by numbers in the KRNL column that range from 40% to 100%.

Figure 11–2 Processor Utilization by Process

```
                                      Processor Utilization by Process

Process Name    Samples  %        Total Time System Time System Processor Utilization by Access Mode
                                  (Seconds) (Seconds) Time % KRNL EXEC SUPR USER CMPT IPL>0 IPL>2
--------------- ---- -----        -------- -------- -----  ----  ---  ---  ---  --- ----  ----
Interrupt Stack  8497  4.72%      84.97     84.97    100%  100%   0%   0%   0%   0%   0% 100%
NULL CPU TIME   51786 28.77%     517.86    517.86    100%  100%   0%   0%   0%   0%   0% 100%
SWAPPER            77  0.04%       0.77      0.73     95%  100%   0%   0%   0%   0% 100%  96%
SYMBIONT_0023      14  0.01%       0.14      0.13     93%   79%  14%   0%   7%   0%  71%  57%
SYMBIONT_0024      19  0.01%       0.19      0.18     95%   84%  11%   0%   5%   0%  84%  84%
ERRFMT             18  0.01%       0.18      0.12     67%   94%   6%   0%   0%   0%  50%  33%
CACHE_SERVER        0  0.00%       0.00      0.00      0%    0%   0%   0%   0%   0%   0%   0%
CLUSTER_SERVER      0  0.00%       0.00      0.00      0%    0%   0%   0%   0%   0%   0%   0%
OPCOM              25  0.01%       0.25      0.24     96%   96%   0%   0%   4%   0%  64%  64%
JOB_CONTROL      1123  0.62%      11.23      9.49     85%   66%  19%   0%  15%   0%  59%  52%
CONFIGURE          71  0.04%       0.71      0.65     92%   96%   4%   0%   0%   0%  95%  58%
SMISERVER           4  0.00%       0.04      0.04    100%   75%  25%   0%   0%   0%  75%  50%
SERVER_002E        10  0.01%       0.10      0.10    100%   90%  10%   0%   0%   0%  90%  90%
USERA           33258 18.48%     332.58    221.17     67%   59%   7%   0%  34%   0%  55%  45%
NETACP            556  0.31%       5.56      4.06     73%  100%   0%   0%   0%   0%  69%  65%
USERU             157  0.09%       1.57      0.86     55%   39%  15%  45%   0%   0%  36%  34%
REMACP              4  0.00%       0.04      0.04    100%  100%   0%   0%   0%   0% 100%  75%
SPM_CAPACITY     1392  0.77%      13.92     10.77     77%   99%   1%   0%   0%   0%  93%  92%
EMACS VTA6        459  0.26%       4.59      2.09     46%   44%   6%   0%  50%   0%  34%  26%
SYMBIONT_0001       0  0.00%       0.00      0.00      0%    0%   0%   0%   0%   0%   0%   0%
USERZ             956  0.53%       9.56      7.35     77%   87%   9%   0%   3%   0%  61%  46%
SERVER_0024         2  0.00%       0.02      0.02    100%  100%   0%   0%   0%   0% 100% 100%
USERY               0  0.00%       0.00      0.00      0%    0%   0%   0%   0%   0%   0%   0%
USERX               0  0.00%       0.00      0.00      0%    0%   0%   0%   0%   0%   0%   0%
DNS$ADVER         233  0.13%       2.33      1.69     73%   94%   0%   0%   6%   0%  55%  49%
DFS$COM_ACP         1  0.00%       0.01      0.01    100%  100%   0%   0%   0%   0%   0%   0%
DFS$00010001_1      0  0.00%       0.00      0.00      0%    0%   0%   0%   0%   0%   0%   0%
NMAIL_0001          0  0.00%       0.00      0.00      0%    0%   0%   0%   0%   0%   0%   0%
SYMBIONT_0009      32  0.02%       0.32      0.31     97%   88%   9%   0%   3%   0%  85%  66%
SYMBIONT_0010      18  0.01%       0.18      0.18    100%   94%   0%   0%   6%   0%  89%  56%
SYMBIONT_0011      17  0.01%       0.17      0.17    100%   82%  18%   0%   0%   0%  82%  76%
SYMBIONT_0012      16  0.01%       0.16      0.16    100%  100%   0%   0%   0%   0% 100%  81%
MSCPmount         728  0.40%       7.28      3.75     52%   33%  22%  45%   0%   0%  28%  22%
USERQ              33  0.02%       0.33      0.24     73%   67%  15%  18%   0%   0%  54%  33%
USERC           18769 10.43%     187.69     67.44     36%   33%   4%   0%  63%   0%  31%  28%
SERVER_0004         5  0.00%       0.05      0.05    100%  100%   0%   0%   0%   0% 100%  80%
PID=20A000D9    45828 25.46%     458.28     91.25     19%   15%   5%  10%  75%   0%  63%  65%
PID=20A000DD     3260  1.81%      32.60     24.64     76%   67%   9%   3%  20%   0%  62%  53%
PID=20A000DC     2624  1.46%      26.24     20.10     77%   68%  10%   4%  18%   0%  63%  53%
```

12 Evaluating Performance Using Graphic Displays

This chapter provides the following information about the VAX SPM Graphic Display utilities:

- Overview of the Graphic Display utilities
- Description of the RESOURCE command
- Description of the INVESTIGATE command
- Performance evaluation using Investigate graphic displays

12.1 Overview of the Video Graphic Display Utilities

The VAX SPM Graphic Display utilities collect performance data on a VAX/VMS system and displays its output to a video terminal. Some displays are available on terminals that support DEC_CRT characteristics, such as the VT100. Use the SET TERMINAL/DEC_CRT command to set charactistics for these terminals. Other displays are available only for ReGIS-compatible terminals, such as the VT240. If the terminal supports color, or if an external color monitor is attached, a multicolored display will be generated. The display can also be printed on a graphics dot matrix printer.

12.1.1 Prerequisites

The VAX SPM Video Graphics utilities have the following mandatory and optional software and hardware requirements:

Mandatory Requirements

- For real-time mode, the following privileges are required:

  ```
  CMKRNL        TMPMBX        PSWAPM        WORLD
  ```

  The SPMTIMER driver must also be loaded.
- For playback mode, TMPMBX is required. The SPMTIMER driver need not be loaded.
- A ReGIS-compatible terminal such as the VT125, VT240, VT241, VT330, VT340, or PC350 is needed for most displays invoked through INVESTIGATE.
- A terminal with DEC_CRT characteristics, such as VT100, is needed for displays invoked through RESOURCE.
- The Graphic Display utilities take commands from SYS$COMMAND or from files pointed to by logical names SPM$INI_DISPLAY_INVESTIGATE and SPM$INI_DISPLAY_RESOURCE, and writes output to SYS$OUTPUT. SYS$COMMAND must be assigned to a terminal-type device.

Optional Requirements

- A color monitor
- A graphics printer, such as the LA34-VA or LA50

Any number of users with these resources can simultaneously run the VAX SPM Video Graphics utilities.

12.1.2 The Graphic Display Commands

There are two commands used to invoke VAX SPM graphic displays from the DCL command level. They are:

```
$ PERFORMANCE DISPLAY=RESOURCE
            and
$ PERFORMANCE DISPLAY=INVESTIGATE
```

The two commands provide different functions and capabilities as shown in Table 12–1.

Table 12–1 Differences Between Resource and Investigate Utilities

Resource	Investigate
Requires DEC_CRT terminal support	Requires ReGIS-compatible terminal
Displays data for up to eight nodes on a VAXcluster system, or data for up to eight disks	Displays data for a single node
Does not support playback mode	Displays data from a log file in playback mode
	Has an EXPLAIN command to show an explanation of displays

The Resource and Investigate displays have the following in common:

- Both are invoked in a similar manner and display data in real-time mode.
- Both support the use of initialization files.
- Both permit you to specify the types of resources for which data is collected.
- Both permit you to specify the types of resources for which data is displayed.

Each of these similarities is described in the following sections.

12.1.2.1 Invoking the Graphic Displays

The video display commands have the following format:

```
$ PERFORMANCE DISPLAY=[ RESOURCE    ]  [node-name,...]
                      [ INVESTIGATE ]
```

These commands accept a parameter (which is a list of node names) and several qualifiers. If a node is not given, the local node is assumed. An asterisk (*) for the node name initiates collections on all nodes in a VAXcluster system. You can invoke video displays on nodes that are not part of a VAXcluster system as described in Chapter 22.

These commands invoke the displays in interactive mode, which allows you to monitor the performance of your running system.

During a display, you can type subcommands to invoke other displays, access HELP, or exit the displays. For example, type CTRL/Z or EXIT to end the display and return to the DCL prompt. Type HELP to learn more about the display commands. When you type a character, the display is interrupted and the prompt SPM> and your input appear at the bottom left of the screen.

12.1.2.2 Using an Initialization File

An initialization file contains a series of VAX SPM subcommands. These subcommands establish nondefault settings for types of data and characteristics of the graphic displays.

There are two ways to execute the commands in an initialization file: by defining a logical name and by using the /COMMAND qualifier.

- In the case of the DISPLAY_RESOURCE command, the logical name SPM$INI_DISPLAY_RESOURCE can be defined to point to the initialization file containing RESOURCE subcommands. Likewise, for the INVESTIGATE command, the logical name SPM$INI_DISPLAY_INVESTIGATE points to another initialization file containing INVESTIGATE subcommands. Commands in the initialization file are executed when the matching display command is given:

  ```
  $ PERFORMANCE DISPLAY=[ RESOURCE    ]
                        [ INVESTIGATE ]
  ```

  If an initialization file is not defined or if it does not translate to a valid file specification, the default settings are used.

- Use the /COMMAND qualifier in the command line to specify an initialization file other than the one pointed to by the logical name SPM$INI_DISPLAY_RESOURCE or SPM$INI_DISPLAY_INVESTIGATE. The following command is an example:

  ```
  $ PERFORMANCE DISPLAY= [ RESOURCE    ] /COMMAND=VTDISP.INI
                         [ INVESTIGATE ]
  ```

  The above command executes the VAX SPM subcommands in VTDISP.INI, which control initial screen display and set-up characteristics. Since no node name is given, the display is for the local node. If an invalid file name is given, a fatal error occurs.

  Use the /NOCOMMAND qualifier to inhibit the processing of an initialization file as in the following command:

  ```
  $ PERFORMANCE DISPLAY=[ RESOURCE    ] /NOCOMMAND
                        [ INVESTIGATE ]
  ```

An initialization file is not processed even if the logical name SPM$INI_DISPLAY_RESOURCE or SPM$INI_DISPLAY_INVESTIGATE translates to a valid file specification.

If the /COMMAND qualifier is omitted, a search is made for the logical name:

SPM$INI_DISPLAY_RESOURCE in the case of a RESOURCE command
or
SPM$INI_DISPLAY_INVESTIGATE in the case of an INVESTIGATE command.

If the logical name is not defined or does not translate to a valid file specification, the default settings are used.

If /COMMAND is given without a file specification, an error results.

12.1.2.3 Specifying Resources for Which Data is Collected

The RESOURCE and INVESTIGATE commands collect default types of data. If you wish to collect data other than the default, you need to specify this data using qualifiers when you invoke the graphic displays.

For the DISPLAY_RESOURCE command, the /CLASS and /DISK qualifiers control the classes and disks for which data is collected and displayed. For the INVESTIGATE command, there are the /CLASS, /DISK, and /DEVICE qualifiers.

These qualifiers can be used both globally and locally on the command line to affect the types of resources for which data is collected. VAX SPM supports global and local qualifiers in the same way that VMS supports them. Table 12–2 describes the use of global and local qualifiers.

Table 12–2 Global and Local Qualifiers

Global Qualifier	Local Qualifier
Not associated with a node in a command line	Associated with a node in a command line
Affects all nodes in a command line except those with local qualifiers that override the global qualifier	Affects only its associated node and overrides any global qualifiers for that node

The following command illustrates the use of global and local qualifiers:

```
$ PERFORMANCE DISPLAY=INVESTIGATE/CLASS=DEVICE/DISK=(DUA0,DUA1)-
GREEN/DEVICE=X,BLUE,PURPLE/DISK=DU3
```

In the previous example, the qualifiers /CLASS=DEVICE and /DISK=(DUA0,DUA1) are used globally, and the qualifiers /DISK=DU3 and /DEVICE=X are used locally. The following explains how these qualifiers affect data collection for nodes in this command:

- Node GREEN has the local qualifier /DEVICE=X, which overrides the effect of the global qualifier /CLASS=DEVICE. Node GREEN is also affected by the global qualifier /DISK=(DUA0,DUA1). Therefore, data will be collected for only device X and for disks DUA0 and DUA1 for node GREEN.
- Node BLUE has no local qualifiers and is affected by the global qualifiers /DISK=(DUA0,DUA1) and /CLASS=DEVICE. Therefore, data will be collected for disks DUA0 and DUA1 and all devices for node BLUE.
- Node PURPLE has the local qualifier /DISK=DU3 and is affected by the global qualifier /CLASS=DEVICE. The local qualifier overrides the global qualifier /DISK=(DUA0,DUA1). Therefore, data will be collected for disk DU3 and all devices on node PURPLE.

In the following command, real-time display is initiated for the local node. Only disks YELLOW$DUA0 and HSC001$DUA1 are present in displays for disk statistics.

```
$ PERFORMANCE DISPLAY=INVESTIGATE/DISK=(YELLOW$DUA0,HSC001$DUA1)
```

Certain disks can be selected by specifying them with /DISK. Devices will be included in the displays only if they are specified using /CLASS=ALL or /CLASS=DEVICE together with /DEVICE to indicate a list of specific devices. Consider the following example:

```
$ PERFORMANCE DISPLAY=INVESTIGATE GREEN/DEVICE=X
```

The /CLASS qualifier is missing from the command line; therefore, data will not be collected for device X or for any other devices on node GREEN.

12.1.2.4 Specifying Resources for Which Data is Displayed

In addition to collecting default data, the RESOURCE and INVESTIGATE commands also display default data. To select specific data for display, you can use the DEFINE GROUP and SET GROUP subcommands. The DEFINE GROUP subcommand allows you to define a group of resources for which you want data displayed. The SET GROUP command displays data for this group. You can use the DEFINE GROUP command while viewing a Resource or Investigate display, or include it in an initialization file. Once a group is defined, it can be displayed by typing the SET GROUP subcommand.

For example, you may want to define disks used by a particular group of users on your system by using commands similar to the ones in the following example:

```
DEFINE GROUP/DISK CADCAM_USERS $255$DUA0,$255$DUA1,$255$DUA2,$255$DUA3,
$255$DUS1,$255$DUS2,$255$DUS3,$255$DUS4.

DEFINE GROUP/DISK ALL-IN-ONE_USERS $255$DUA13,$255$DUA15,
$255$DUA16,$255$DUA17,$255$DUS6,$255$DUS7,$255$DUS8,$255$DUS9
```

If your VAXcluster system contains both VAX 8800 and MicroVAX II systems, you may wish to define groups of large and small CPUs for display. The commands in the following example could be typed into an initialization file or in response to the video display SPM> prompt:

```
DEFINE GROUP/NODES LARGE_CPUS GREEN,PURPLE,RED
DEFINE GROUP/NODES SMALL_CPUS AQUA,VIOLET,PINK
```

A command to bring up a screen other than the default one can be included in the initialization file. For example, suppose that the logical name SPM$INI_DISPLAY_INVESTIGATE translates to a file that contains the single line:

```
DISPLAY CPU
```

When the command below is typed, the CPU display for the local node is shown for ReGIS-compatible terminals. For non-ReGIS terminals, the Load Balance is the only valid Investigate display; therefore, no display is shown.

```
$ PERFORMANCE DISPLAY=INVESTIGATE
```

12.2 The RESOURCE Command

The purpose of the Resource series of graphic displays is to permit evaluation of resource utilization in interactive mode for one or more nodes in a VAXcluster system. Graphic displays can also be invoked for a single node that is not a member of a VAXcluster system. All displays can be accessed by using a terminal with DEC_CRT characteristics, such as the VT100.

Use the command below to begin collecting data for all nodes in a VAXcluster system and to display memory metrics in real-time mode for up to eight nodes.

```
$ PERFORMANCE DISPLAY=RESOURCE *
```

Table 12–3 lists RESOURCE qualifiers.

Table 12–3 PERFORMANCE DISPLAY=RESOURCE Command Syntax

Command Format

PERFORMANCE DISPLAY=RESOURCE [node-name, . . .]

Command Qualifier	Default
/CLASS=(item[, . . .])	/CLASS=DISK
/[NO]COMMAND[=file-spec]	Not present in command line
/DISK=([disk, . . .])	/DISK
/VERSION	Not present in command line

12.2.1 Subcommand Syntax for DISPLAY=RESOURCE

Once a Resource display has been shown, subcommands can be given to control the display and its characteristics. These commands can also be included in an initialization file as described in Section 12.1.2.2. Each command consists of one or more keywords. Some keywords take a value.

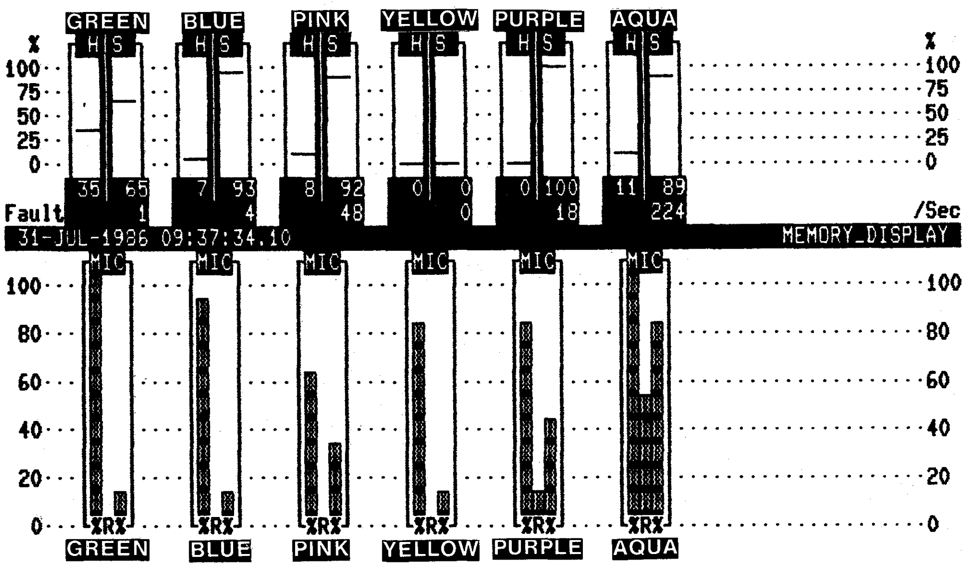
Table 12–4 shows the RESOURCE commands given as subcommands at the VAX SPM level. These commands control display characteristics such as the type of display, the viewing time or time that the display remains static after being shown, and the disks and nodes that are seen in the display "window." The DISPLAY subcommand controls which of the three possible resource displays (memory, disk, and CPU) appear on the screen. Figures Figure 12–1 through Figure 12–3 show examples of each type of display.

Table 12–4 RESOURCE Subcommands

Subcommand	Meaning
DEFINE GROUP group-name item-list /DISK /NODE	Define groups of disks and nodes for display.
DISPLAY CPU_DISPLAY DISK_DISPLAY MEMORY_DISPLAY	Select desired display.
EXIT or CTRL/Z	Exit the display.
HELP	Display help text for subcommands.
READ BROADCAST ERROR	Read broadcast or error message.
SET GROUP group-name /DISK /NODE	Set disk or node group for display.
SET VIEWING_TIME n	Set display update and data collection interval to be n seconds.
SHOW COLLECTION GROUP VIEWING_TIME	Show nodes and disks for collection, defined node and disk groups, and current update interval.
SPAWN	Spawn a subprocess.
CTRL/W	Refresh the display.

Other RESOURCE subcommands perform functions such as defining groups of disks and nodes for display, returning help information, reading broadcast messages, and spawning a subprocess. For example, the SET GROUP subcommand causes a group of disks or nodes to be selected as the current group for display. For a complete description of these subcommands, see the *VAX SPM Reference Manual.*

Figure 12–1 Memory Resource Display



ZK-5698-86

12.2.1.1 RESOURCE Subcommand Defaults

The default characteristics and display are as follows:

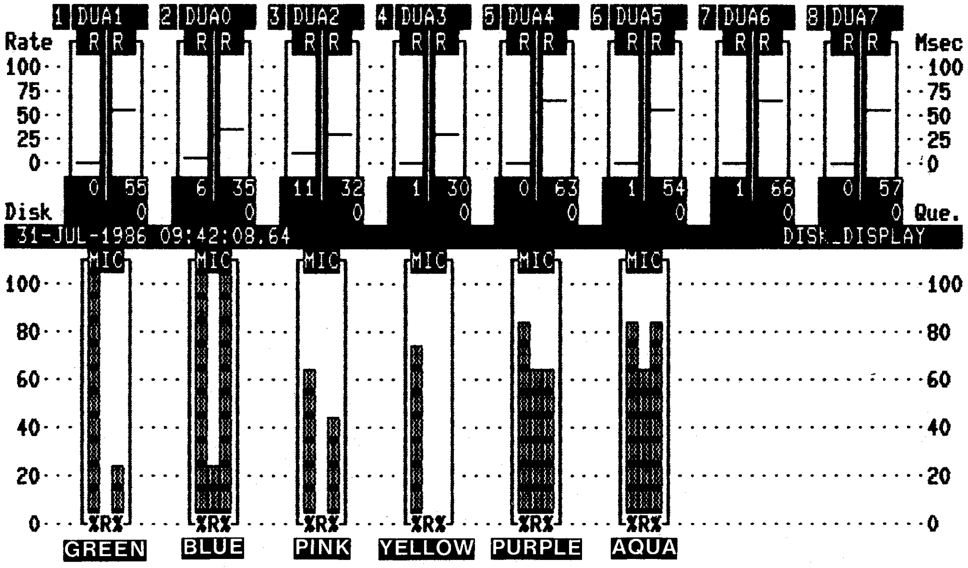
```
SET VIEWING_TIME=10
DISPLAY MEMORY_DISPLAY
```

The viewing time is in seconds.

There is a default group consisting of nodes or disks first shown when the Resource displays are entered. To return to the default nodes, type this command:

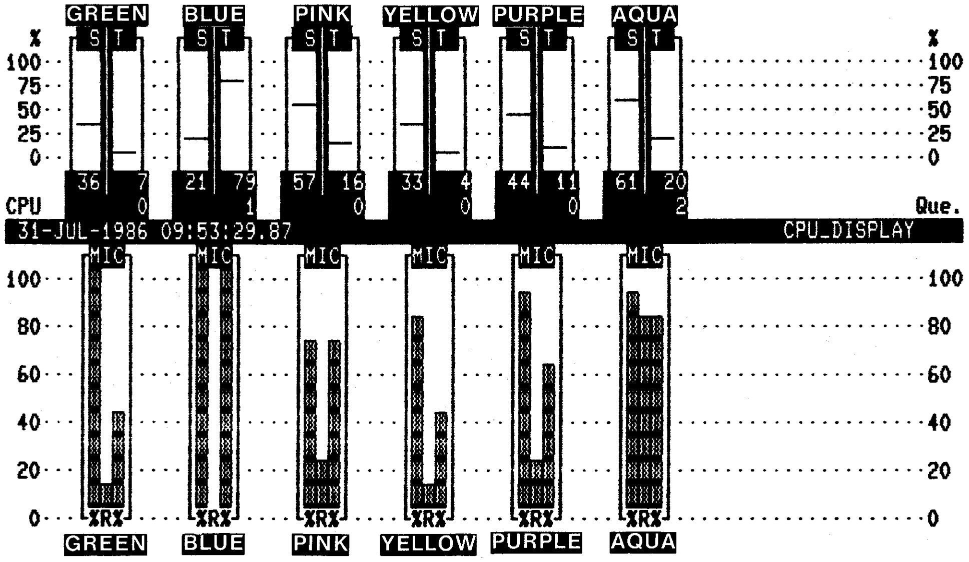
```
SET GROUP/NODE  DEFAULT
```

Figure 12–2 Disk Resource Display



ZK-5699-86

Figure 12–3 CPU Resource Display



ZK-5700-86

12.2.2 Balancing VAXcluster System Utilization Using the Resource Video Display

Use the Resource graphic displays to determine the workload on nodes and disks on your VAXcluster system, and balance the workload as necessary. For example, if your Resource display shows one node that has a high percentage of a resource in use, you may wish to move work from that node to other nodes and thereby balance resource utilization on your VAXcluster system.

12.3 The INVESTIGATE Command

The purpose of the Investigate series of graphic displays is help you evaluate performance on one node. You can evaluate performance interactively or in playback mode. All displays except one require use of a ReGIS-compatible terminal. Table 12–5 lists INVESTIGATE qualifiers.

Table 12–5 PERFORMANCE DISPLAY=INVESTIGATE Command Syntax

Command Format

```
PERFORMANCE DISPLAY=INVESTIGATE [node-name, . . . ]
```

Command Qualifier	Default
/BEFORE[=time]	/BEFORE=end of file
/CLASS=(item[, . . .])	/CLASS=DISK
/[NO]COLOR	/COLOR
/[NO]COMMAND[=file-spec]	Not present in command line
/DEVICE=([device, . . .])	/DEVICE
/DISK=([disk, . . .])	/DISK
/INPUT=file-spec	Not present in command line
/[NO]PLAYBACK	/NOPLAYBACK
/SINCE[=time]	/SINCE=beginning of file
/VERSION	Not present in command line

Use the command below to display system metrics interactively for the local node. If the terminal is ReGIS-compatible and no initialization file is present, the System Overview display appears on the terminal screen. For a non-ReGIS terminal, the Load Balance display appears.

```
$ PERFORMANCE DISPLAY=INVESTIGATE
```

12.3.1 Invoking INVESTIGATE in Playback Mode

Use the /PLAYBACK qualifier on the command line to specify that data comes from a log or history file.

The /BEFORE and /SINCE qualifiers control which portion of the log file or history file will be displayed. Only data with a timestamp within the range specified by the /BEFORE and /SINCE qualifiers is played back.

The /INPUT qualifier can be used globally or locally to give the log or history file specification. Specified globally, it indicates a log file or a history file. When the /INPUT qualifier is specified locally with a node name, it indicates the log history file is to be used when displaying data for the particular node.

If a node name is not given, the node in the log file or the first node in the history file is assumed.

Type the following command to begin a video display for node GREEN in playback mode:

```
$ PERFORMANCE DISPLAY=INVESTIGATE/PLAYBACK-
 GREEN/INPUT=GREENAPR86.LOG
```

12.3.2 Subcommand Syntax for DISPLAY=INVESTIGATE

Once an Investigate display has appeared, use subcommands to control the display and its characteristics. These commands can also be included in an initialization file. Each command consists of one or more keywords. Some keywords take a value. Two main groups of commands are the SET and DISPLAY commands. In addition, there are miscellaneous commands.

12.3.2.1 INVESTIGATE SET Subcommand

The SET commands given as subcommands at the VAX SPM level are shown in Table 12–6. Each command begins with the keyword SET. These commands control display characteristics such as the node for display, the scaling factors for display graphs, the disks, the devices being displayed, the terminal type, and the viewing time or time that the display remains static.

For example, the SET GROUP subcommand causes a group (defined using DEFINE GROUP) of devices, disks, or servers to be selected for display. For a complete description of these subcommands, see the *VAX SPM Reference Manual*.

Table 12–6 INVESTIGATE SET Subcommands

Subcommand	Meaning
SET WINDOW/PRIORITY= REAL_TIME TIME_SHARING	Set window to show real-time or time-sharing process priorities.
SET WINDOW/PROCESS= TOPCPU TOPDIO TOPBIO TOPFAULT TOPWORKING_SET	Display processes according to criterion selected.
SET WINDOW/IO= DISKS SERVERS	Set window to show disks or servers in I/O statistics.
SET DISPLAY_TYPE REGIS ANSI	Set terminal type.
SET LEGEND/NOLEGEND	Turn legend on and off.
SET SCALING n /PROCESS /WORKING_SET /RATE_PER_SECOND	Change scale (number within a tick mark) for process, working set size, and I/Os per second scales.
SET NODE nodename	Specify name of node for display.
SET GROUP group-name /DEVICE /DISK /SERVER /NODE nodename	Specify a group of devices, disks, or servers for display.
SET VIEWING_TIME n	Specify viewing time, or time after which display will be updated.

12.3.2.2 INVESTIGATE DISPLAY Subcommands

The DISPLAY subcommands are shown in Table 12–7. Each command begins with the keyword DISPLAY. These commands control the type of display.

The Load Balance display is the only Investigate display available for both ReGIS and DEC_CRT terminals. It contains the same information in both cases. On ReGIS terminals it is a Kiviat graph, and for DEC_CRT terminals it is a form of bar graph. The System Overview is the default display for ReGIS-compatible terminals, and for non-ReGIS or DEC_CRT terminals the default is the Load Balance display. Figure 12–5 is an example of the ReGIS version of the System Overview Display. Figure 12–6 is an example of the ReGIS Load Balance display (Kiviat) and Figure 12–7 is an example of the DEC_CRT (ANSI) version of the Load Balance display. The other displays show memory, I/O, and CPU statistics and require a ReGIS-compatible terminal. Figure 12–10 is an example of the Memory display, Figure 12–12 is an example of the I/O display, and Figure 12–14 is an example of the CPU display.

Table 12–7 INVESTIGATE DISPLAY Subcommands

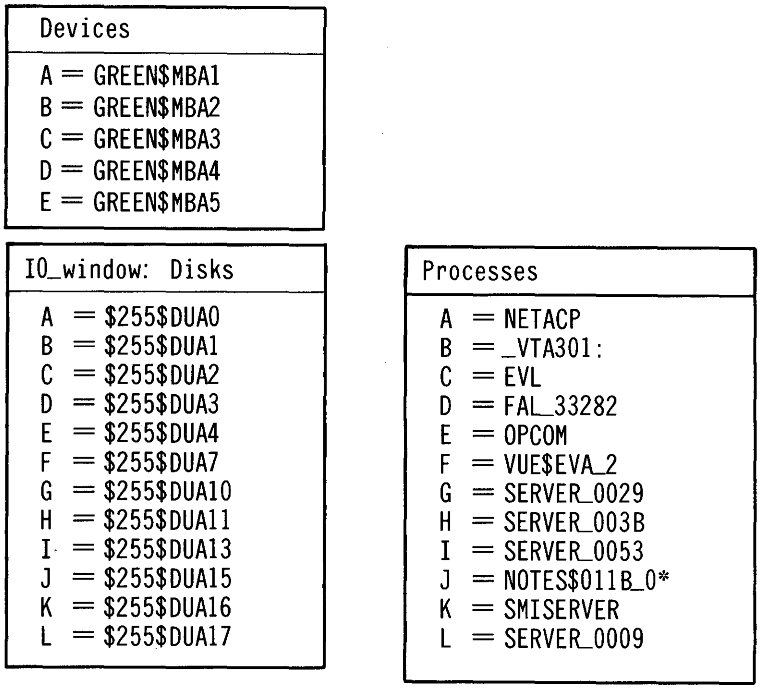
Subcommand	Meaning
DISPLAY CPU_DISPLAY	Display CPU statistics
DISPLAY IO_DISPLAY	Display I/O statistics
DISPLAY LOAD_BALANCE_DISPLAY	Display load balance statistics
DISPLAY MEMORY_DISPLAY	Display memory statistics
DISPLAY SYSTEM_OVERVIEW_DISPLAY	Display system overview statistics

12.3.2.3 INVESTIGATE Miscellaneous Subcommands

The miscellaneous subcommands are shown in Table 12–8. These commands perform functions such as defining groups of devices, disks, and servers for display, putting up a screen of text that explains a given display, returning help information, printing the current screen on a graphics printer, reading broadcast messages, and spawning a subprocess.

For example, the PRINT subcommand causes the current display to print on a printer attached to a graphics terminal. The /LEGEND qualifier used with PRINT prints the display and a legend page showing disk names, process names, and so on. An example of a legend page is shown in Figure 12–4.

Figure 12–4 Example Legend from PRINT/LEGEND



```
Devices
A = GREEN$MBA1
B = GREEN$MBA2
C = GREEN$MBA3
D = GREEN$MBA4
E = GREEN$MBA5
```

```
IO_window: Disks          Processes
A  = $255$DUA0            A  = NETACP
B  = $255$DUA1            B  = _VTA301:
C  = $255$DUA2            C  = EVL
D  = $255$DUA3            D  = FAL_33282
E  = $255$DUA4            E  = OPCOM
F  = $255$DUA7            F  = VUE$EVA_2
G  = $255$DUA10           G  = SERVER_0029
H  = $255$DUA11           H  = SERVER_003B
I  = $255$DUA13           I  = SERVER_0053
J  = $255$DUA15           J  = NOTES$011B_0*
K  = $255$DUA16           K  = SMISERVER
L  = $255$DUA17           L  = SERVER_0009
```

Table 12–8 INVESTIGATE Miscellaneous Subcommands

Subcommand	Meaning
DEFINE GROUP group-name item-list /DEVICE /DISK /SERVER /NODE=nodename	Define groups of devices, disks, and servers for display.
EXIT or CTRL/Z	Exit the display.
EXPLAIN	Show explanation of display.
HELP	Display HELP text for subcommands.
PRINT /DOUBLE_SIZE /LEGEND	Print display, large size (optional) and with legend.
READ BROADCAST ERROR	Read broadcast or error message.
SHOW COLLECTION GROUP VIEWING_TIME	Show nodes and devices for collection; defined disk, device, and server groups; and current update interval.
SPAWN	Spawn a subprocess.

12.3.2.4 INVESTIGATE Subcommand Defaults

The default characteristics and display are as follows:

```
SET WINDOW/PRIORITY=TIME_SHARING
SET WINDOW/PROCESS=TOPCPU
SET WINDOW/IO=DISKS
SET DISPLAY_TYPE REGIS
SET LEGEND
SET SCALING/PROCESS 2
SET SCALING/WORKING_SET 10
SET SCALING/RATE_PER_SECOND 10
SET VIEWING_TIME 10
DISPLAY SYSTEM_OVERVIEW
```

The viewing time is in seconds.

12.4 Evaluating Performance Using Investigate Graphic Displays

The Investigate displays are designed for system tuning. The System and Load Balance displays provide enough information to determine which of the three main system resources (memory, I/O, or CPU) is a limitation. To investigate a limitation in more detail, the memory, I/O, and CPU displays are provided.

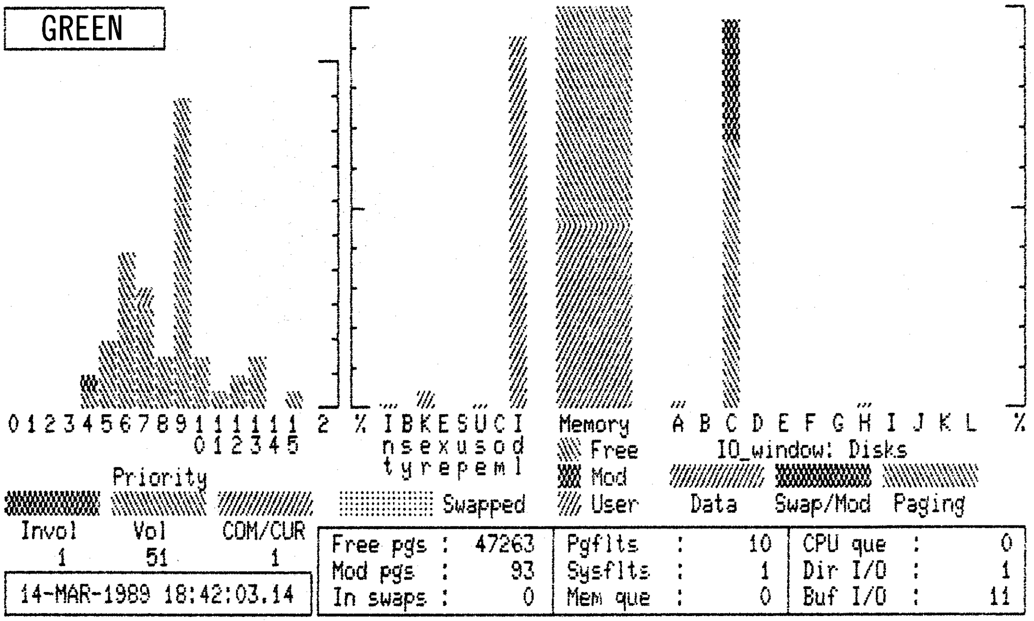
Note: **For a detailed description of a system tuning methodology, see the *Guide to VAX/VMS Performance Management*.**

A system tuning methodology using VAX SPM is provided by the Explain screens that accompany the Investigate displays. Typing EXPLAIN when viewing a given display returns a screen of text indicating how the information in the display can be used to determine a limited resource and to isolate the cause of the limitation.

12.4.1 System Overview Display

Begin your tuning investigation by using the System Overview (Figure 12-5) or Load Balance displays (Figures 12-6 and 12-7).

Figure 12–5 ReGIS System Overview Display



```
Invol     Vol     COM/CUR
  1        51        1
14-MAR-1989 18:42:03.14

Free pgs :  47263   Pgflts   :    10   CPU que  :    0
Mod pgs  :     93   Sysflts  :     1   Dir I/O  :    1
In swaps :      0   Mem que  :     0   Buf I/O  :   11
```

The Explain screens for these displays are shown in Figure 12–8 and Figure 12–9. When determining a limitation in a main resource, investigate the resources in this order:

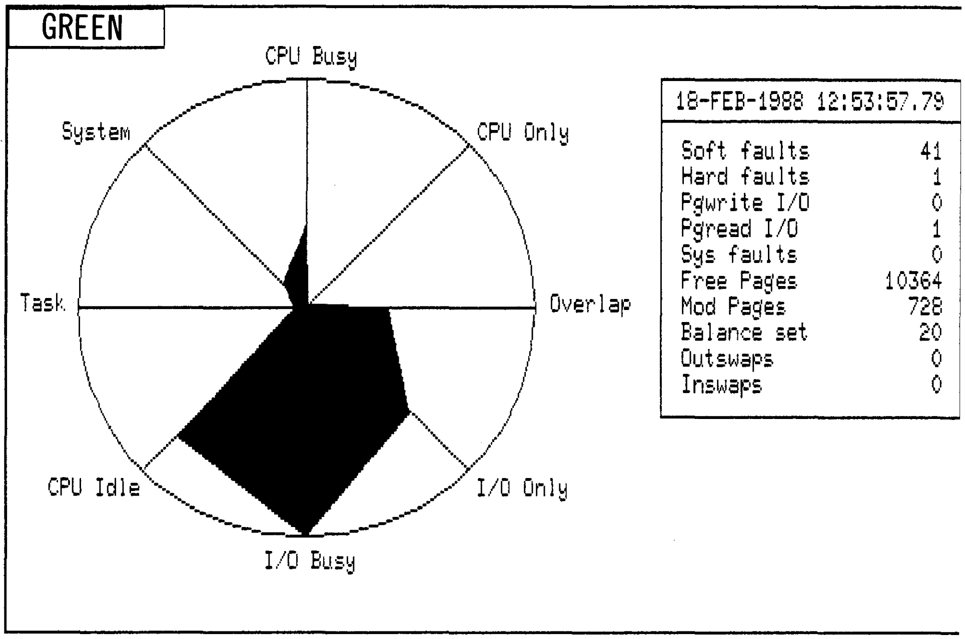
1 Memory

2 I/O

3 CPU

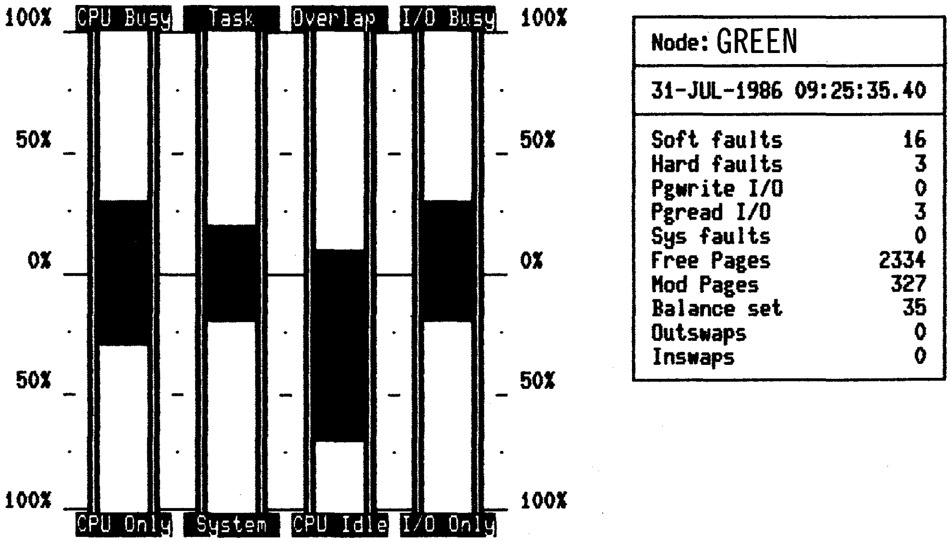
The order is important because memory limitations cause paging and swapping, which lead to I/O and CPU problems.

Figure 12–6 System Load Balance Display (ReGIS)



```
18-FEB-1988 12:53:57.79

Soft faults         41
Hard faults          1
Pgwrite I/O          0
Pgread I/O           1
Sys faults           0
Free Pages       10364
Mod Pages          728
Balance set         20
Outswaps             0
Inswaps              0
```

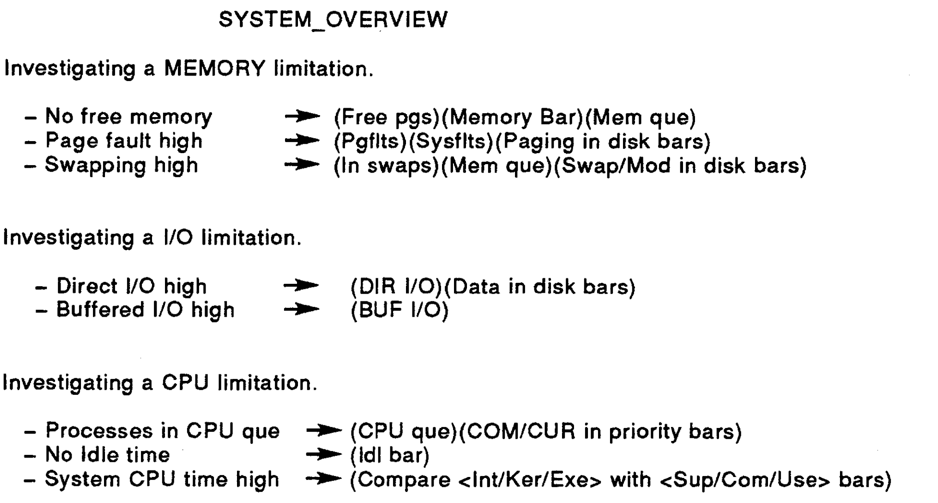
Figure 12–7 LOAD_BALANCE Display (ANSI)



```
Node: GREEN

31-JUL-1986 09:25:35.40

Soft faults         16
Hard faults          3
Pgwrite I/O          0
Pgread I/O           3
Sys faults           0
Free Pages        2334
Mod Pages          327
Balance set         35
Outswaps             0
Inswaps              0
```

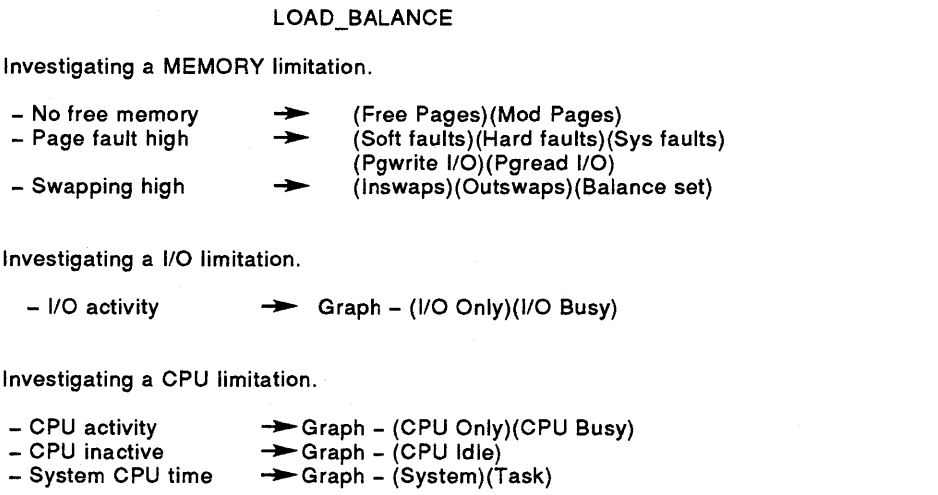
ZK-5697-86

Figure 12–8 Explain Screen for System Overview Display



```
                    SYSTEM_OVERVIEW

Investigating a MEMORY limitation.

  - No free memory        ➤ (Free pgs)(Memory Bar)(Mem que)
  - Page fault high       ➤ (Pgflts)(Sysflts)(Paging in disk bars)
  - Swapping high         ➤ (In swaps)(Mem que)(Swap/Mod in disk bars)

Investigating a I/O limitation.

   - Direct I/O high      ➤   (DIR I/O)(Data in disk bars)
   - Buffered I/O high    ➤   (BUF I/O)

Investigating a CPU limitation.

  - Processes in CPU que  ➤ (CPU que)(COM/CUR in priority bars)
  - No Idle time          ➤ (Idl bar)
  - System CPU time high  ➤ (Compare <Int/Ker/Exe> with <Sup/Com/Use> bars)
```

Figure 12–9 Explain Screen for Load Balance Display



```
                    LOAD_BALANCE

Investigating a MEMORY limitation.

 - No free memory      ➤     (Free Pages)(Mod Pages)
 - Page fault high     ➤     (Soft faults)(Hard faults)(Sys faults)
                             (Pgwrite I/O)(Pgread I/O)
 - Swapping high       ➤     (Inswaps)(Outswaps)(Balance set)

Investigating a I/O limitation.

  - I/O activity       ➤  Graph - (I/O Only)(I/O Busy)

Investigating a CPU limitation.

 - CPU activity        ➤Graph - (CPU Only)(CPU Busy)
 - CPU inactive        ➤Graph - (CPU Idle)
 - System CPU time     ➤Graph - (System)(Task)
```

12.4.1.1 Investigating a Memory Limitation

The chief indicators of a memory limitation are:

- No free memory

  Look at the value of free pages, the memory bar histogram, and the memory queue (Mem que).

  If the value of free pages approaches that of FREELIM (the system parameter that sets a lower limit on the number of pages on the free list), this indicates a memory shortage.

  The memory bar histogram shows the amount of memory used by the free and modified page lists with respect to that available for user working sets. Again, a shortage of memory for the page caches (free plus modify lists) indicates a high degree of user memory utilization, and consequently a memory shortage.

  A nonzero value of "Mem que" indicates processes in the memory queue in computable outswapped states, awaiting memory that is unavailable.

- Page fault high

  Look at the value of Pgflts (page faults), Sysflts (system faults), and paging in the IO_window for disks.

  For a VAX-11/780 CPU, a value of Pgflts [page fault rate, or number of faults (hard and soft, including system) per second] greater than 100 is cause for concern. For other CPUs see the Table 7–2, CPU Factor by Processor and multiply the factor for your processor by 100. If Pgflts is greater than this number, page faults may be excessive on your system.

  Sysflts (the rate at which pages are faulted into the system working set) should be no more than 1 fault per second; otherwise, the system is faulting itself to do work on the users' behalf. If system faults are high, it may be necessary to increase the value of system parameter SYSMWCNT, which controls the system working set size.

  If disks are spending an excessive percentage of their time doing paging (indicated by the disk bars in the IO_window), then a memory limitation is causing harmful I/O effects.

- Swapping high

  Look at the value of "Inswaps," the value of "Mem que," and the amount of swapping and modified page writing done by the disks (Swap/Mod in the disk bars).

  Inswaps gives the number of processes that were swapped back into memory during the last sample interval, and Mem que is the number of processes waiting for memory. If these are significant, then swapping is a problem on your system, and either indicates a memory limitation or memory management problem.

  If disks are spending an excessive percentage of their time doing swapping and modified page writing (indicated by the disk bars in the IO_window), then a memory limitation is causing harmful I/O effects.

12.4.1.2 Investigating an I/O Limitation

The chief indicators of an I/O limitation are:

- Direct I/O high

  Look at the value of Dir I/O, as well as Data in the disk bar histogram.

  Dir I/O gives the system-wide direct I/O rate, including all disks. If there is a high direct I/O rate for your system, the disks may be a bottleneck. Use Table 7–2 to find the factor for your CPU and multiply it by 30. A value for direct I/O rate that exceeds this number indicates a high direct I/O rate.

  Data in the disk bars shows how much of the disk's time is spent doing I/O on behalf of the user. A high rate for a given disk may indicate that that disk is a bottleneck.

- Buffered I/O high

  A high value of Buf I/O (the buffered I/O rate) may indicate an I/O limitation. Use Table 7–2 to find the factor for your CPU and multiply it by 100. If the value of Buf I/O exceeds this number, a high Buffered I/O rate is indicated.

12.4.1.3 Investigating a CPU Limitation

The chief indicators of a CPU limitation are:

- Processes in the CPU queue

  Look at the value of CPU queue and at processes in the COM/CUR states in the Priority bars.

  CPU que gives the number of processes waiting for the CPU (COM state). There is a real queue if this value is greater than 1 (the process displaced by VAX SPM; the NULL process is not counted). The existence of a real CPU queue indicates a CPU limitation.

  A significant number of processes in the COM/CUR states (COM = computable, waiting for the CPU, CUR = the current process) also indicates a CPU queue, and consequently a CPU limitation.

- No Idle time

  Look at the Idle (Idl) bar in the center of the display. If there is no CPU idle time, then CPU is a limitation.

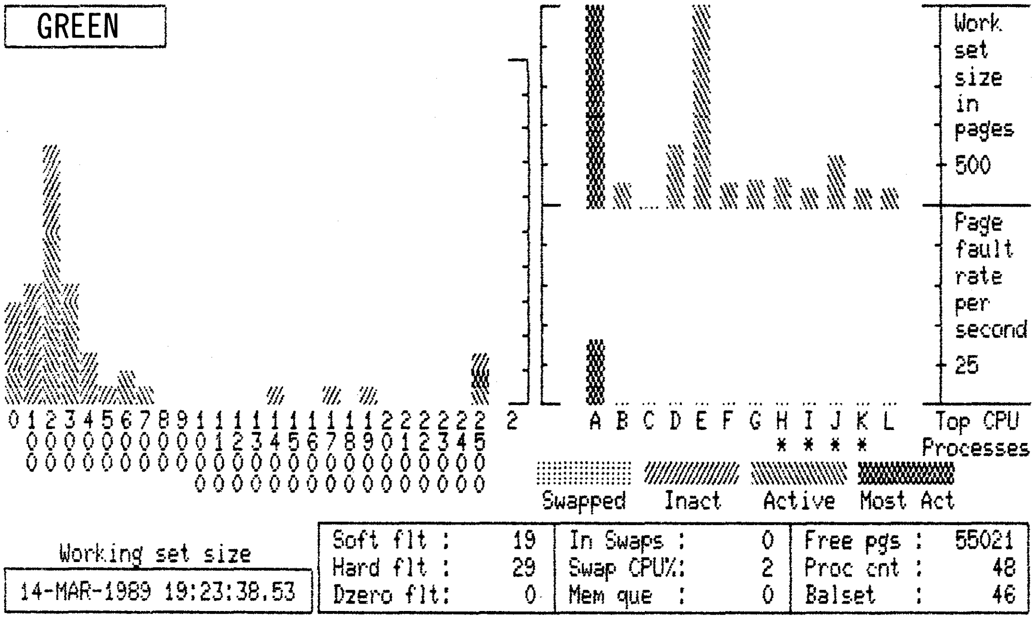
- System CPU time high

  Compare the System CPU time (sum of Int/Ker/Exe bars) with the Task CPU time (sum of Sup/Com/Use bars). The system CPU time is the sum of time spent on the interrupt stack, and in kernel and executive modes. The task CPU time is the sum of time spent in supervisor, compatibility, and user modes. The sum of interrupt and kernel CPU time should not exceed 40% in most environments.

12.4.2 Isolating the Cause of a Memory Limitation

If an examination of the system overview (as indicated above) reveals a memory limitation, the cause of the limitation can be investigated in more detail using the Memory display. Figure 12–10 is an example of the Memory display.

Figure 12–10 Memory Display



```
Working set size      | Soft flt :   19 | In Swaps :   0 | Free pgs :  55021
                      | Hard flt :   29 | Swap CPU%:   2 | Proc cnt :     48
14-MAR-1989 19:23:38.53 | Dzero flt:  0 | Mem que  :   0 | Balset   :     46
```

The Explain screen for the Memory display (Figure 12–11) draws attention to the following indicators:

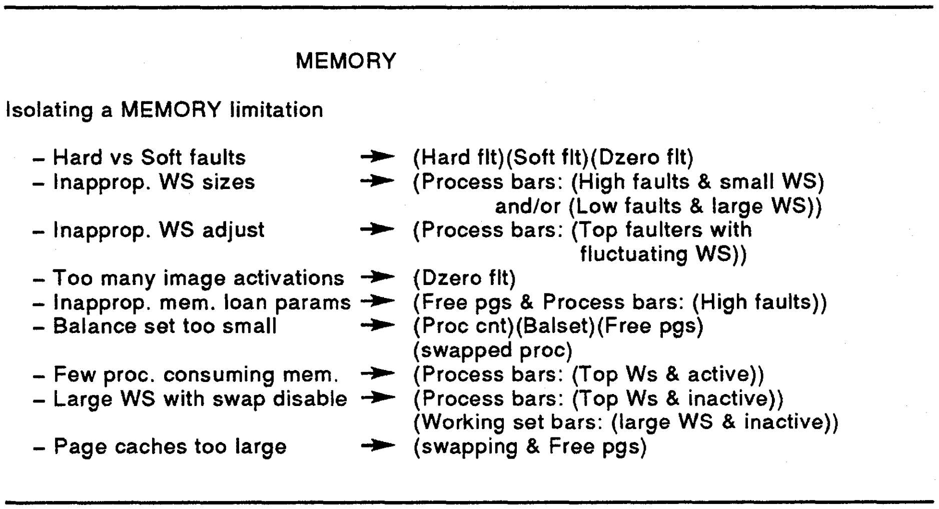
- Hard vs Soft faults

  Look at the value of Hard flt and Soft flt. Hard flt gives the number of page faults per second that were resolved by reading from the disk. Soft flt gives the number of faults per second resolved from memory. A hard fault involves I/O and is more expensive than a soft fault. Hard faults in a properly managed system should be no more than about 10% of the total faults (Hard flt + Soft flt).

- Inappropriate working set (WS) sizes

  Look at the Process bars at the right on the Memory display. This shows the working set size and page fault rate for the top faulting processes. Adjust the scaling factors, if necessary. Look for processes that are faulting heavily but have small working sets. If your system has ample memory, increase WSQUOTA and WSEXTENT for these processes. If memory is short on your system, increase WSQUOTA and WSEXTENT for these processes at the expense of processes that are not faulting but have large working sets.

Figure 12–11 Explain Screen for Memory Display



```
                           MEMORY

Isolating a MEMORY limitation

  - Hard vs Soft faults          →  (Hard flt)(Soft flt)(Dzero flt)
  - Inapprop. WS sizes           →  (Process bars: (High faults & small WS)
                                          and/or (Low faults & large WS))
  - Inapprop. WS adjust          →  (Process bars: (Top faulters with
                                                  fluctuating WS))
  - Too many image activations   →  (Dzero flt)
  - Inapprop. mem. loan params   →  (Free pgs & Process bars: (High faults))
  - Balance set too small        →  (Proc cnt)(Balset)(Free pgs)
                                    (swapped proc)
  - Few proc. consuming mem.     →  (Process bars: (Top Ws & active))
  - Large WS with swap disable   →  (Process bars: (Top Ws & inactive))
                                    (Working set bars: (large WS & inactive))
  - Page caches too large        →  (swapping & Free pgs)
```

- Inappropriate automatic working set adjustment (AWSA) parameters

  Look at the Process bars at the right on the Memory display. Look for top faulting processes with fluctuating working set sizes. If the working set size for such a process increases and decreases accompanied by page faulting, then the AWSA parameters may be out of adjustment. System parameters that affect automatic working set adjustment are PFRATH, PFRATL, WSINC, WSDEC, AWSTIME, AWSMIN, GROWLIM, BORROWLIM, and QUANTUM. Automatic decrementing can be turned off by setting PFRATL = 0 (this is normally recommended). Do not change any of the other parameters without a thorough understanding of the AWSA mechanism.

- Too many image activations

  Look at the value of Dzero flts. A large number of demand zero faults indicates an excessive number of image activations. Activating an image in a process involves considerable overhead. If Dzero faults is a large percentage of total faults (Hard flt + Soft flt), image activations may be excessive. Paging induced by image activations is unlikely to respond to system parameter changes. Application design changes are needed.

- Inappropriate memory loan parameters

  Look at the value of Free pgs (free memory in pages) and at the page fault rate (Hard flt + Soft flt, and process fault rates in the Process bars). A large amount of free memory accompanied by high page fault rates is one indication of inappropriate memory loan parameters.

  This could occur because: BORROWLIM is too high, and you are not loaning memory when you could be; PFRATH may be too high, so that processes are forced into high fault rates before they are given additional memory; and WSEXTENT may be too low, so that processes reach their limits and are still faulting heavily.

  Another indication of inappropriate memory loan parameters is a lot of processes in their working set extent regions when other processes are being swapped. Look at the Process bars for these processes (high working set sizes accompanied by swapping). If BORROWLIM and/or GROWLIM are too generous, you may be giving away memory when you should not be doing so.

- Balance set too small

  Look at Proc cnt (number of processes on system), Balset (number of processes in balance set), Free pgs (number of pages of free memory), and swapped processes. If the balance set count is too small, processes will be swapped even if there is still free memory. If Balset is significantly less than Proc cnt, and Free pgs is adequate, then the balance set count is too low. Set the system parameter BALSETCNT to a value two less than the system parameter MAXPROCESSCNT.

- A few active processes consuming memory

  Look at the Process bars, in particular for active processes with large working sets. For example, a low priority compute-bound process is less likely to be swapped than one that performs terminal I/O. They may cause other processes to swap.

  Decreasing DORMANTWAIT may help if the large processes are above their working set quotas. You can also suspend the large process with SET PROCESS/SUSPEND and allow the swapper to trim it back to SWPOUTPGCNT. The underlying problem may be that WSQUOTA is too large for the process.

- Large processes with swapping disabled

  Look at Working Set and Process bars for inactive processes with large working sets. If these processes have swapping disabled, they cannot be swapped but retain memory at the expense of other processes. Use the system dump analyzer (SDA) to see if a large, inactive process has the PSWAPM (prohibit swap mode) bit set.

- Inappropriate page cache sizes

  Look at the page fault rate (Hard flt and Soft flt), free memory (Free pgs), and swapping (Working Set and Process bars).

  If the overall fault rate is high, and the faults are mostly soft faults, the page cache may be too large. This may also be accompanied by swapping and extensive free and modified page lists. The page cache is encroaching on memory that could be made available for working sets.

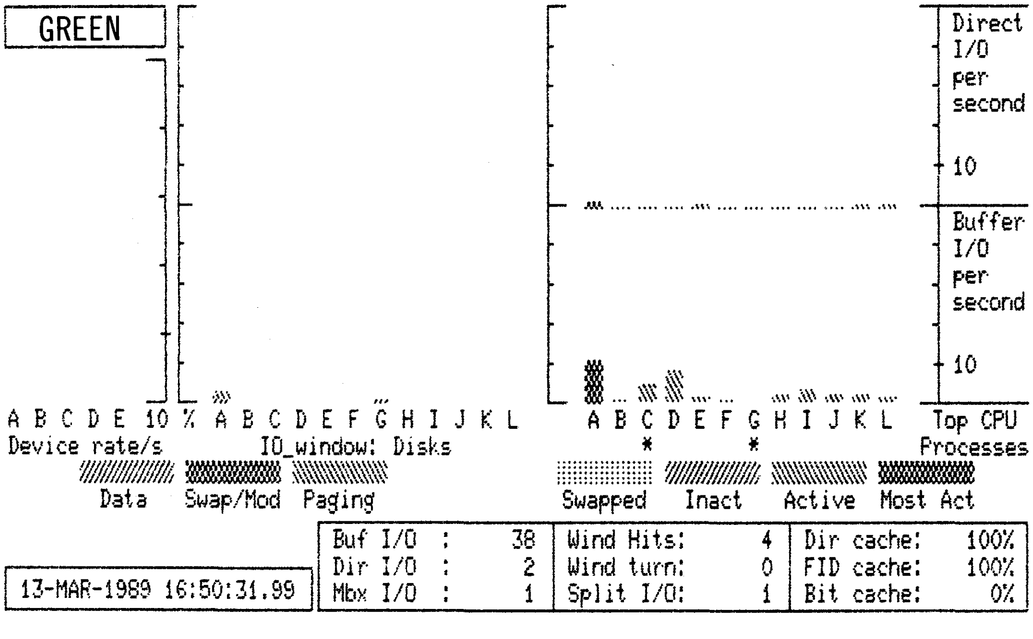
  If the overall faulting rate is low while the hard fault rate is high, the page cache is ineffective; that is, the free page list and/or modified page list is too small. There is ample memory for working sets but the caching effectiveness is low.

  The sizes of the page caches are controlled by the system parameters FREELIM, FREEGOAL, MPW_LOLIMIT, and MPW_THRESH.

12.4.3 Isolating the Cause of an I/O Limitation

If an examination of the system overview (indicated above) reveals an I/O limitation, the cause of the limitation can be investigated in more detail using the I/O display. Figure 12–12 is an example of the I/O display.

Figure 12–12 I/O Display



```
13-MAR-1989 16:50:31.99

Buf I/O  :      38   Wind Hits:     4   Dir cache:    100%
Dir I/O  :       2   Wind turn:     0   FID cache:    100%
Mbx I/O  :       1   Split I/O:     1   Bit cache:      0%
```

The Explain screen for the I/O display (Figure 12–13) draws attention to the following indicators:

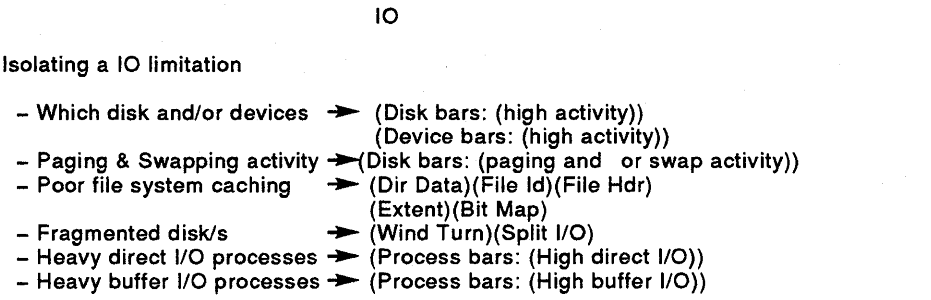
- Which disk and/or devices

  Look for the most active disks and devices in the disk and device bars at left on the I/O display. Disks perform direct I/O and devices such as terminals and line printers perform buffered I/O. Compare the I/O rates observed with the rated capacity of the device. A rate of 16 to 25 I/Os per second per disk is a moderate to heavy load for RA-series and MASSBUS disks, and more than 25 I/Os per second is a heavy load.

- Paging and swapping activity

  Disk activity may be due to paging and/or swapping. Look at the disk bars at left, and note how much of a high disk I/O rate is due to Paging and Swapping/Modified page writing. If this is significant, it indicates a memory shortage or management problem that is causing an I/O limitation. Having an additional paging file on another disk might help this situation.

Figure 12–13 Explain Screen for I/O Display



```
                              IO

Isolating a IO limitation

 - Which disk and/or devices  ➤  (Disk bars: (high activity))
                                 (Device bars: (high activity))
 - Paging & Swapping activity ➤(Disk bars: (paging and  or swap activity))
 - Poor file system caching   ➤ (Dir Data)(File Id)(File Hdr)
                                (Extent)(Bit Map)
 - Fragmented disk/s          ➤ (Wind Turn)(Split I/O)
 - Heavy direct I/O processes ➤ (Process bars: (High direct I/O))
 - Heavy buffer I/O processes ➤ (Process bars: (High buffer I/O))
```

- Poor file system caching

  Look at the values of Dir cache, FID cache, and Bit cache. These indicate cache effectiveness for the Directory FCB, File Id, and Bitmap caches. Metrics for additional caches are given as part of tabular reports. The attempt rate on a cache is the number of times per second an attempt was made to find data in the cache. The cache effectiveness is the percentage of attempts during which data was actually found (hit rate). The effectiveness should be 75% or greater for the active caches; otherwise, the caches are too small. If the attempt rate on a cache is small, however, an ineffective cache may not be serious. See the tabular reports for cache attempt rates.

  If a cache is heavily used but ineffective, consider increasing the cache size. Relevant system parameters are ACP_HDRCACHE (File header), ACP_MAPCACHE (Bitmap cache), and so on.

- Fragmented disk(s)

  A fragmented file has more than one extent, where each extent is a contiguous series of disk blocks. Look at window turns and split I/Os (Wind turn and Split I/O). On a well-run system, they should be no greater than 0.1. A window turn means that a file was so fragmented, an I/O request caused the XQP to read another set of extent pointers into the WCB (window control block). By default, a WCB contains up to 7 extent pointers. A split I/O means that more than one extent pointer was used to satisfy a logical I/O request.

  If Wind turn and Split I/O indicate disk fragmentation, use PERFORMANCE REPORT=DISK_SPACE to further analyze the problem as described in Chapter 10.

- Heavy direct I/O processes

  Look at the Process bars for processes with a high direct I/O rate. The process may be executing a program that employs explicit specification of Direct QIOs but the program may not be properly designed. Program redesign may include caching to improve efficiency of QIOs.

- Heavy buffered I/O processes

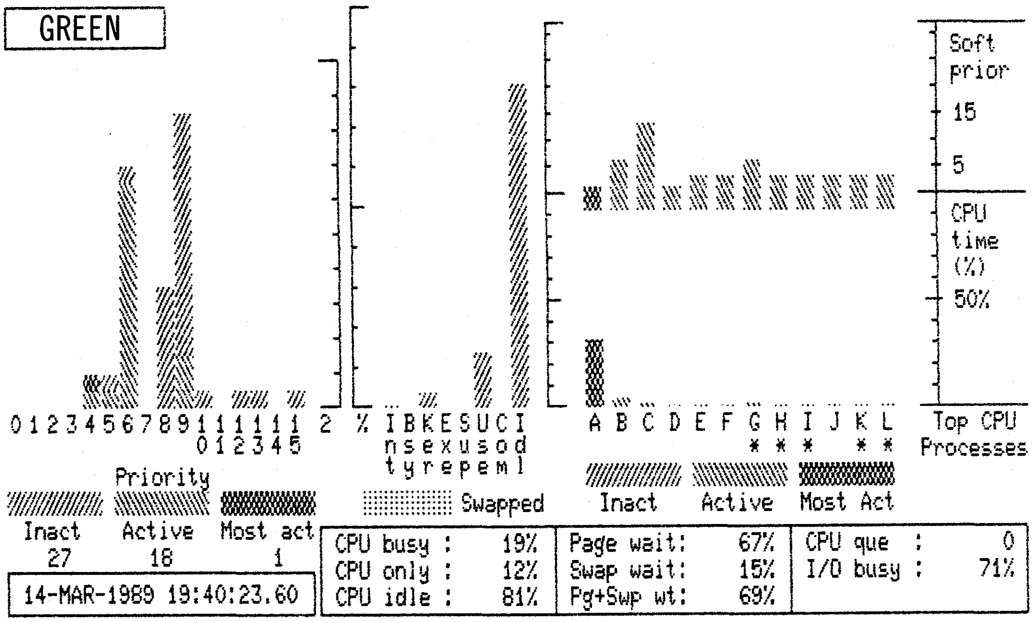
  Look at the Process bars for processes with a high buffered I/O rate. Investigate the application (use of device) to determine whether the high buffered I/O rate is really necessary. Look at Device rate/s to determine the device.

  A high buffered I/O rate with processes in the COM state suggests that the CPU is constricted by terminal I/O demands, although other devices may be responsible.

12.4.4 Isolating the Cause of a CPU Limitation

If an examination of the system overview (as indicated above) reveals a CPU limitation, the cause of the limitation can be investigated in more detail using the CPU display. Figure 12–14 is an example of the CPU display.

Figure 12–14 CPU Display



```
14-MAR-1989 19:40:23.60

CPU busy :   19%   Page wait:    67%   CPU que  :     0
CPU only :   12%   Swap wait:    15%   I/O busy :   71%
CPU idle :   81%   Pg+Swp wt:    69%
```

The Explain screen for the CPU display (Figure 12–15) draws attention to the following indicators:

- Any available CPU

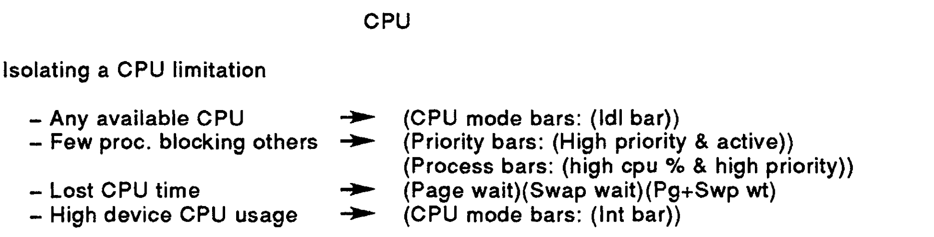
  Look at the Idl bar (CPU Idle time) in the CPU mode bars in the middle of the CPU display. If there is no idle time, the CPU is a bottleneck.

- A few processes blocking other processes

  The blocking high-priority process may be: running an inefficient program, acting as a server, or acting as a process with which other processes must communicate.

  Look at the Priority bars for high-priority, active processes, or at the Process bars for high-priority processes with a high CPU time %. Corrective action might include changing process priorities in the user authorization file, defining priorities in the user login command file, or changing the priorities of processes while they execute.

Figure 12–15 Explain Screen for CPU Display



```
                                   CPU

Isolating a CPU limitation

   - Any available CPU         ➤  (CPU mode bars: (Idl bar))
   - Few proc. blocking others ➤  (Priority bars: (High priority & active))
                                  (Process bars: (high cpu % & high priority))
   - Lost CPU time             ➤  (Page wait)(Swap wait)(Pg+Swp wt)
   - High device CPU usage     ➤  (CPU mode bars: (Int bar))
```

- Lost CPU time

  Look at Page wait, Swap wait, and Pg+Swp Wt. CPU time may be "lost" because the CPU has to wait for disk transfers, or page or swap I/O to complete.

  A high value of Pg+Swp Wt is cause for concern. It indicates a memory problem resulting in a CPU limitation.

- High device CPU usage

  Look at the CPU mode bars, Int (interrupt stack). A high value for Int may be cause for concern. Processes may be blocked from using the CPU because of too many device interrupts.

  Use PERFORMANCE COLLECT=SYSTEM_PC to collect system-wide PC samples and determine the system module usage (for example, the device driver); hence, the device(s) responsible for the excessive interrupts.

- Excessive kernel and/or executive CPU time

  Look at the Ker (Kernel) and Exe (Executive) bars in the CPU mode bars. If time in Kernel mode mode is excessive and is not due to page faulting, or if time in Executive mode is excessive, use PERFORMANCE COLLECT=SYSTEM_PC to collect system-wide PC samples, and determine the processes and system modules responsible.

  Interrupt plus Kernel CPU time should not be greater than 40% of total CPU time.

13 Using the Analyzer

This chapter describes the VAX SPM Analyzer and how to use the Analyzer in the following ways:

- Getting started using the Analyzer
- Invoking the Analyzer to run interactively
- Invoking the Analyzer to run noninteractively
- Requesting information during a session
- Directing a session
- Analyzing VAXcluster system performance
- Changing the values of the performance indicators that the Analyzer uses
- Modifying the value of disk and CPU scaling factors that the Analyzer uses
- Using an initialization file to customize a session

This chapter provides examples of how to use the Analyzer interactive commands and qualifiers during a session. Examples of Analyzer responses are provided to show how the Analyzer works. The wording of these examples may differ from the responses you see. The *VAX SPM Reference Manual* provides a detailed description of the ANALYZE interactive and noninteractive commands.

13.1 Getting Started Using the Analyzer

The Analyzer is an automated version of the hands-on approach to system tuning described in Chapter 7 and Chapter 8. It reads a VAX SPM log file and evaluates the resources of memory, I/O, and CPU to determine if one of these resources is limiting. When it finds a limiting resource, it investigates the factors contributing to the limitation and makes recommendations for correcting it. After you have made the correction, run the Analyzer on new data to verify that the limitation has been corrected.

The Analyzer can benefit both new and experienced performance analysts. It may be used any time collections are running and can be used on an open log file to provide a quick analysis of current system performance during the work day. It can be used on a closed log file to analyze performance at the end of a collection period.

New performance analysts learn the hands-on methodology by observing the Analyzer as it evaluates performance and by requesting additional information at each test.

Experienced performance analysts can use the Analyzer to confirm the results of a hands-on performance investigation or to uncover performance problems they may have overlooked.

Note: **Care must be taken when evaluating the results and applying recommendations. It is not appropriate to make changes to a system based upon a nonrecurring problem, no matter how dramatic the problem might be.**

A performance bottleneck may come and go, or change over time. Therefore, the Analyzer may come up with different or even contradictory recommendations for different time periods. The user must decide whether or not to follow the recommendations. If the same recommendation appears consistently, then the recommendation should be heeded.

13.1.1 Prerequisites

The Analyzer can read data only from a VAX SPM log file because history files do not contain the necessary configuration and process information. The following classes of data must be present in the log file:

```
CPU                 Hardware Configuration  Process
Device              I/O                     Scheduler
Disk                Memory                  System Parameters
File Primitives     Page Fault              XQP Cache
```

Use the /CLASS=ANALYZE qualifier with the PERFORMANCE COLLECT command to collect the required classes of data. The following is an example:

```
$ PERFORMANCE COLLECT=[ TUNE     ] /CLASS=ANALYZE
                      [ CAPACITY ]
```

13.1.2 The ANALYZE Command

The ANALYZE command has the following format:

```
$ PERFORMANCE ANALYZE=LOG_FILE [log-file specification,...]
```

This command accepts one parameter, which is the name of the log file or log files to be analyzed, and several qualifiers.

Log files may be specified using any valid file specification including the wild card characters * and %. Specifying one file initiates a single node analysis. Specifying more than one file initiates a VAXcluster system analysis. When initiating a VAXcluster system analysis, specify one file for each node in the VAXcluster system. Be sure that all the log files cover the same period of cluster operation.

The Performance Analyzer may be invoked interactively from the DCL level and noninteractively from a command file by submitting it to the batch queue.

13.2 Running the Analyzer Interactively

You may invoke the Analyzer to run interactively to analyze performance on a single node or a VAXcluster system. The list below describes the flow of a very simple interactive analysis session and the commands that may be used at each point. The sections following the steps describe them.

1 Invoke the Analyzer interactively.

```
$ PERFORMANCE ANALYZE=LOG_FILE
```

2 Invoke Histogram mode to display the graph of log file activity and select a period for anlaysis.

```
HISTOGRAM logfile.dat
SPM> DOWN
SPM> MARK
SPM> DOWN
SPM> READ
```

3 Start the analysis.

```
STEP
```

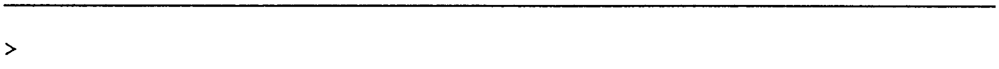
4 End the analysis.

```
EXIT
```

To run the Analyzer interactively, type the ANALYZE command as follows:

```
$ PERFORMANCE ANALYZE=LOG_FILE
```

The Analysis screen is displayed:



```
>
```

When the screen displays, you can type interactive commands to select data and proceed with the analysis.

The Analyzer interactive commands are organized into three groups according to the types of tasks they perform: start-up, control, and advisory. Table 13–1 lists the Analyzer commands according to these groups.

Table 13–1 Analyzer Interactive Commands

Command	Description
	Start-Up Command
SET CLUSTER_FILES	Defines a set of log files that describe a cluster for analysis
HISTOGRAM	Provides an interactive method for selecting intervals for analysis
READ	Loads data from a specified log file for analysis
SET DISK_FACTOR	Modifies the value of the disk factor used to scale threshold values

Table 13–1 (Cont.) Analyzer Interactive Commands

Command	Description
	Start-Up Command
SET CPU_FACTOR	Modifies the value of the CPU factor that is used to scale threshold values
SET TIME	Specifies the analysis interval used when reading log file data
SET THRESHOLD	Modifies the value of a threshold used in subsequent tests
SAVE	Writes a file containing all threshold values modified during a session
@	Executes a file containing values for threshold and session characteristics that customize your analysis session
SET PAUSE	Sets the value of the session characteristic that determines the wait between tests when a session proceeds from the GO command

Command	Description
	Control Command
STEP	Runs one test and stops for user interaction
GO	Runs the analysis without waiting for interaction between tests
BACK	Erases the result of the current test and returns to the previous test
CHANGE	Selects the alternate result of the current test
RESTART	Sets the analysis back to the first test
EXIT	Ends an analysis session
QUIT	Ends an analysis session without displaying a report reminder
ATTACH	Enables switching control from the current process to another process within the current job
SPAWN	Suspends the current analysis process and creates a VAX/VMS subprocess
REFRESH	Redraws the current screen

Table 13–1 (Cont.) Analyzer Interactive Commands

Command	Description
	Advisory Command
DIRECTORY	Displays a listing of the specified directory
HELP or ?	Displays information on the ANALYZE commands
CURRENT	Shows the current test and result
REVIEW	Displays a list of all tests and their results
MORE	Displays in-depth information on the current test
SHOW CLUSTER_FILES	Displays the log files that describe the currently defined cluster
SHOW DATAFILE	Displays the name and other information for the current log file
SHOW DISK_FACTOR	Displays the value of the disk factor that is used to scale threshold values
SHOW CPU_FACTOR	Displays the value of the CPU factor that is used to scale threshold values
SHOW PAUSE	Displays the value of the pause session characteristic that determines the display time between tests when the GO command is selected
SHOW THRESHOLD	Displays the current value of a threshold
SHOW TIME	Displays the current value of the /SINCE and /BEFORE times used when reading log file data
REPORT	Creates a report of the current analysis session

13.2.1 Specifying a Log File and Analysis Interval

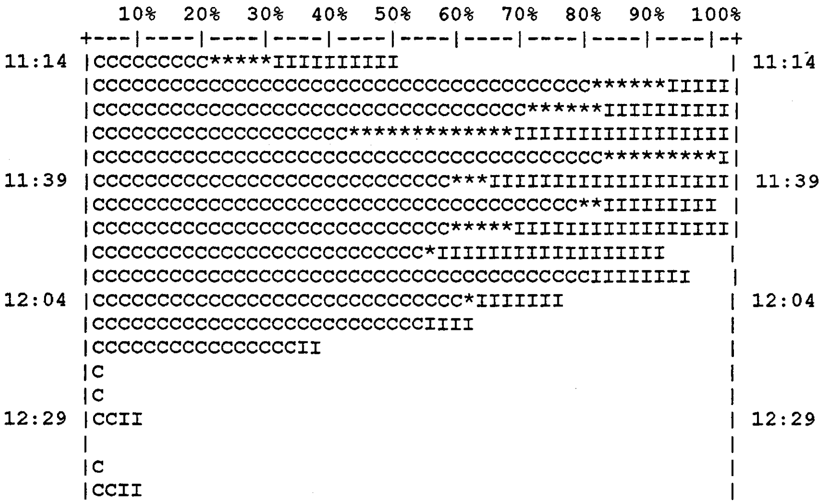
This section describes how to interactively select a period for analysis from a log file using the HISTOGRAM commands.

At the VAX SPM display, use the following command to specify the name of the log file you want to analyze and to display the histogram graph:

```
> HISTOGRAM SPM$COLLECT_TUNE.DAT
```

A graph similar to the VAX SPM System Summary graph displays on the screen. This graph provides a view of system activity and resource use for the entire collection period. Figure 13–1 is an example.

Figure 13–1 Histogram Display



```
         10%  20%  30%  40%  50%  60%  70%  80%  90%  100%
      +---|----|----|----|----|----|----|----|----|----|-+
11:14 |CCCCCCCCC*****IIIIIIIIII                          | 11:14
      |CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC******IIIII|
      |CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC******IIIIIIIIII|
      |CCCCCCCCCCCCCCCCCCCC*************IIIIIIIIIIIIIIIII|
      |CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC*********I|
11:39 |CCCCCCCCCCCCCCCCCCCCCCCCCCCC***IIIIIIIIIIIIIIIIIII| 11:39
      |CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC**IIIIIIIII |
      |CCCCCCCCCCCCCCCCCCCCCCCCCCCC*****IIIIIIIIIIIIIIIII|
      |CCCCCCCCCCCCCCCCCCCCCCCCCC*IIIIIIIIIIIIIIIIII     |
      |CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCIIIIIIII   |
12:04 |CCCCCCCCCCCCCCCCCCCCCCCCCCCCC*IIIIIII             | 12:04
      |CCCCCCCCCCCCCCCCCCCCCCCCCCIIII                    |
      |CCCCCCCCCCCCCCCCII                                |
      |C                                                 |
      |C                                                 |
12:29 |CCII                                              | 12:29
      |                                                  |
      |C                                                 |
      |CCII                                              |
```

The scale at the top represents the percentage of resources that were being used during a unit of time. Each horizontal bar represents a unit of time. Periods for which the bars approach the 100% range represent high system activity or hot spots. The bars form a graph comprised of the characters I, C and *. Table 13–2 describes each character.

Table 13–2 Description of Characters in the Histogram Graph

Character	Meaning
C	The percentage of time during the observation period that ONLY the CPU was busy. In multiprocessor configurations, it shows the percentage of time that any of the CPUs are busy.
*	CPU and I/O overlap. The amount of time during the observation period that both a CPU and at least one of the disks for which data is being collected has an operation in progress. It is a positive indicator.
I	The percentage of time that ONLY I/O was being performed. This can be an indication that the CPU had to wait for I/O to complete before useful "work" could be completed. It is usually considered less positive than the above indicators.

The following describes the use of the HISTOGRAM commands when viewing the display and selecting a hot spot for analysis.

A reverse video marker is displayed in the top left-hand corner of the graph on the first interval. The screen scrolls to reveal more of the graph as the marker is moved past the bottom or top of the screen.

- Position the marker down by typing the DOWN command or by pressing Keypad 2.
- Position the marker up by typing the UP command or by pressing Keypad 5.
- Position the marker at a horizontal bar that approaches the 100 percentile.
- Specify the beginning of this period of high system activity for analysis by typing MARK or by pressing Keypad 7. As you move the marker beyond this point, the display is highlighted.
- If you wish to reposition the mark, type UNMARK or press Keypad 9.
- Position the marker down to a horizontal bar that is shorter than previous bars (signifying the end of peak activity) by typing DOWN or pressing Keypad 2.
- Specify the end of the period of high system activity by typing READ. The Analyzer reads the data associated with the period you selected and is ready to begin processing.
- If you do not want to select a hot spot from this log file, type CANCEL to exit from Histogram mode.

Table 13–3 describes the commands and their associated keypad keys for standard and LK201 keyboards. On an LK201 keyboard, you can type commands using the keypad or LK201 keys.

Table 13–3 Histogram Commands and Their Associated Keys

Command	Keypad Key	LK201 Key
DOWN	Keypad 2	
10 DOWN	Keypad 0	Next Screen
READ	^Z	EXIT
MARK	Keypad 7	SELECT
CANCEL		
UNMARK	Keypad 9	
UP	Keypad 5	
10 UP	Keypad .	Prev Screen
HELP	Keypad PF2	HELP

13.2.2 Starting the Analysis

After specifying a log file and selecting an analysis interval, you are ready to start the analysis.

Type the STEP command to run the first test.

```
>STEP
```

After the first test is evaluated, the results are displayed on the screen and the program waits for the next command.

```
Is there sufficient memory?
There is sufficient memory.
>
```

If you want to proceed through the session without interaction between tests, type the GO command.

```
> GO
```

The program proceeds to the end of the session, pausing a few seconds between tests. If you want the program to wait for a command between tests, press the RETURN key during the pause. Type either GO or STEP to continue to the next test.

13.2.3 Ending the Analysis

At the end of the analysis, the recommendation is displayed. Terminate the program and return to the DCL prompt by typing the EXIT command.

```
Recommendation:  If opening and closing of files has been minimized,
      and appropriate file allocations are being used; then adjust
      ACP_HDRCACHE, ACP_MAPCACHE, or ACP_DIRCACHE, use AUTOGEN, and reboot
      the system. See the Guide to VAX/VMS Performance Management for more
      information.
>EXIT
No report has been generated for this session. Do you wish to
generate a report? [Y N]
```

You are given the option of generating a report before exiting the program. Type Y to generate a report before you exit the session. Type N to exit the session without generating a report.

13.3 Running the Analyzer Noninteractively

You may invoke the Analyzer noninteractively when you want only the results of the analysis and not an interactive terminal session.

To run the Analyzer noninteractively, specify the ANALYZE command in a command file and submit the file to the batch queue. The Analyzer analyzes data in the specified log file and produces a report.

The log file parameter must be included in the ANALYZE command for a noninteractive analysis.

The command in the following example is specified in a file called ANALYZE.COM:

```
$ PERFORMANCE ANALYZE=LOG_FILE SPM$COLLECT_TUNE.DAT
```

The command file ANALYZE.COM is submitted to the batch queue by using the following command:

```
$ SUBMIT ANALYZE.COM
```

The analysis session uses default values for all qualifiers. It analyzes data for the entire collection period in the log file SPM$COLLECT_TUNE.DAT and produces a report called ANALYZE.RPT

Table 13–4 lists the qualifiers for the ANALYZE command.

Table 13–4 ANALYZE Command Qualifiers

Command Qualifier	Default
/BEFORE[=time]	/BEFORE=end of file
/COMMAND[=file-spec]	SPM$INI_ANALYZE
/NODE=node-name	None
/OUTPUT[=file-spec]	/OUTPUT=ANALYZE.RPT
/SINCE[=time]	/SINCE=beginning of file
/VERSION	None

Use the /SINCE and /BEFORE qualifiers to specify a hot spot for analysis. The following command specifies times between 2:37 and 4:27 in log file SPM$COLLECT_TUNE.DAT:

```
$ PERFORMANCE ANALYZE=LOG_FILE/SINCE=26-MAY-1988:14:37-
  /BEFORE=26-MAY-1988:16:27 SPM$COLLECT_TUNE.DAT
```

13.4 Requesting Information During a Session

Use the following ANALYZE commands to request information during an interactive analysis session:

- HELP — Displays a description of ANALYZE commands
- MORE — Displays in-depth information on the current test
- DIRECTORY — Displays a specified directory

13.4.1 HELP

Use the HELP command when you want to see a list of all commands or for a description of one command. HELP works in a manner similar to VMS HELP. Type HELP or ? during an interactive analysis session as in the following example:

```
> HELP
```

A list of ANALYZE commands are displayed as shown:

```
Information available:
ATTACH      Authorize       BACK     CHANGE   CURRENT
DIRECTORY   EXIT    GO      HELP     MORE     QUIT     READ
REPORT      RESTART REVIEW  Rules    SAVE     Scaling  SET
SHOW        SPAWN   STEP    Sysgen

Topic?
```

Type the name of the command for which you want information. The MORE command is selected in the following example:

```
MORE

  The MORE command displays additional information about the current
  test.  There are at least three levels of information available for
  each test.  The first level compares the observed data with
  threshold values.  The second describes the source of the
  information analyzed.  The remaining levels provide additional
  in-depth information about the data.  Repeatedly typing the MORE
  command causes the analyzer to display these successive levels.
  Typing MORE after all levels of information have been displayed,
  re-displays the first level.

  Use the MORE command at a pause between tests, when you want to
  learn how the analyzer arrived at a particular conclusion.

Topic?
```

13.4.2 MORE

The MORE command provides additional information about a test. Type MORE while the program has paused between tests to learn how a conclusion was reached about a particular test. In the following example, the value of the data observed (2139 pages of free memory) and the value of the comparison threshold (an expected minimum of 327 pages) are displayed:

```
Is there sufficient memory?
There is sufficient memory.

> MORE

   Free memory:          2139 pages
      Expected: min       327
```

Type the MORE command again to learn the source of the information analyzed as in the following example:

```
> MORE
Data source: VAX SPM report AVE Process-memory counts "Free pages"

Data source: Sysgen parameter FREEGOAL
```

Type the MORE command again to learn additional in-depth information about the test as in the following example:

```
> MORE
FREEGOAL is a system parameter.  If the number of free goals drops below
FREELIM, the swapper will try to get the free list up to FREEGOAL pages
before any borrowing is allowed.
```

When no additional information is available, the following message is displayed:

```
>MORE
No more information for this test.
```

13.4.3 DIRECTORY

Use the DIRECTORY command to display a listing of a directory while running the analyzer. This command supports any standard DCL DIRECTORY qualifiers; however, qualifiers must follow the file specification as shown in the following example:

```
> DIR *.DAT/SIZE

    Directory DISK$:[SPM_USER_A]

       GREEN_1987FEB20.DAT;1
                      1719
       GREEN_1987MAR03.DAT;1
                      1867
       GREEN_1987MAR19.DAT;1
                      1862
       SPM$COLLECT_TUNE.DAT;5
                      1332
       SPM$COLLECT_TUNE.DAT;4
                       199

    Total of 5 files, 6979 blocks.
```

13.5 Directing a Session

There are three ways you can direct an analysis session: you can select the alternate answer to the current test, back up to a previous test, or restart a session at the first test.

The Analyzer provides the CHANGE command to let you select an alternate answer for the current test. Use the CHANGE command to select an alternate answer in order to reject a conclusion you feel is not appropriate for your installation, or to explore an alternate investigation when the performance metrics are very close to the threshold values. The following is an example of the CHANGE command:

```
> STEP

Is swapping excessive?
The in-swap rate is not excessive.

> CHANGE
The next test is now "Memory Investigation: Investigate Swapping"
```

In the above example, even though the Analyzer did not find swapping to be excessive, the user changed this decision and the analysis proceeds to investigate swapping.

The BACK command allows you to move back to previous tests where you can change the decision and explore the alternate investigation. The following is an example using the BACK command:

```
> STEP

The page fault rate is not excessive.

> STEP

The in-swap rate is not excessive.


> BACK
The current test is now:

Is swapping excessive?

The in-swap rate is not excessive.

> CHANGE
```

Use the RESTART command to perform multiple analyses of the same data by using different values for the thresholds (as described in Section 13.7) or changing the results of tests. The RESTART command sets the analysis back to the first test.

In the example below, the analysis session is finished and the recommendation is displayed. Typing the RESTART command sets the analysis back to the first test.

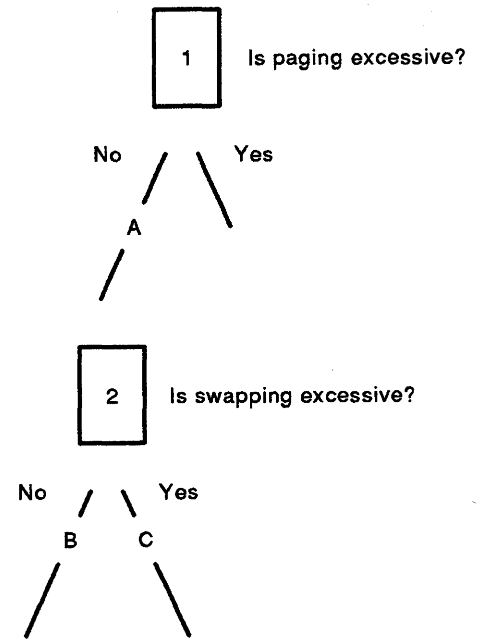
```
Recommendation:  If opening and closing of files has been minimized,
      and appropriate file allocations are being used; then adjust
      ACP_HDRCACHE, ACP_MAPCACHE, or ACP_DIRCACHE, use AUTOGEN, and reboot
      the system. See the Guide to VMS Performance Management for more
      information.
> RESTART
> STEP

Is swapping excessive?
The in-swap rate is not excessive.
```

To fully understand the action of the BACK and CHANGE commands, consider the decision tree shown in Figure 13–2.

While typing commands, you are positioned on the links between decision nodes (represented as boxes). Your current test is the test most recently run relative to your position. This is true even if you have arrived at that position from a BACK command. For example, if the tests for excessive swapping and excessive paging (nodes 1 and 2) have been made and neither is excessive, your position is at B. If you type BACK, you return to position A, ready to run test 2. The screen displays, "The Current test is now: Is swapping excessive"

Note that the effect of the CHANGE command is not permanent; it simply positions you on the alternate link after a decision. For example, if you are at B and you type CHANGE, your position is now at C, and you are ready to follow the alternate path of node 2. If you specify BACK while at C, you will go back to position A, ready to run test 2. Specifying STEP at this point brings you to position B, since the outcome of the test at node 2 (excessive paging) will again be negative.

Figure 13–2 Decision Tree Segment



13.6 Analyzing VAXcluster System Performance

When analyzing VAXcluster system performance, the Analyzer evaluates one node in the same way that it performs a single node analysis. During a cluster analysis, however, it also examines data from other nodes in the cluster to evaluate their influence on the performance of the node being analyzed.

To begin a VAXcluster analysis, use the SET CLUSTER_FILES command to define a set of log files that describe a cluster for analysis. Specify a log file for every node in the VAXcluster system. Do not include any nodes that are not part of the VAXcluster system. Wild card characters may be used, as in the following example:

```
> SET CLUSTER_FILES *_1988SEPT30.DAT
Verifying data files...

> SHOW CLUSTER_FILES
   Node          Log file
   GREEN         GREEN_1988SEPT30.DAT
   BLUE          BLUE_1988SEPT30.DAT
   YELLOW        YELLOW_1988SEPT30.DAT
   AQUA          AQUA_1988SEPT30.DAT
```

A VAXcluster system is described as containing September 30, 1988 log files for nodes GREEN, BLUE, YELLOW, and AQUA.

Once you have defined the cluster, you may use the HISTOGRAM command to view the summary graphs for nodes in the cluster and select periods of high activity for analysis. Type the following command to display the summary graph for the first node GREEN:

```
> HISTOGRAM
```

After you have scrolled through the summary graph (HISTOGRAM commands are described in Section 13.2.1) and noted interesting times, type the following command to terminate the Histogram display for the node.

```
> CANCEL

Reading histogram data from DISK$:[AUSER]GREEN_1988SEP30.DAT ...
No data was read.
```

Continue with the HISTOGRAM/NEXT and CANCEL commands until you have viewed the summary graphs for all nodes in the VAXcluster system. When you have determined a time that you wish to analyze, use the SET TIME command to specify this time as the default time for each node you analyze. For example, if you observed high activity on nodes between 11:30 a.m. and 2:30 p.m., you would use the command in the following example:

```
> SET TIME /SINCE=30-SEP-1988:11:30/BEFORE=30-SEP-1988:14:30
```

You can specify a node for analysis using the READ command. If you want to analyze a particular node first, specify it with the /NODE qualifier as follows:

```
> READ/NODE=GREEN
> STEP
   .
   .
   .
```

When the analysis is finished on node GREEN, you can specify subsequent nodes for analysis using the READ command and the /NEXT qualifier as follows:

```
> READ/NEXT
> STEP
   .
   .
   .
```

After you have analyzed all nodes, you could specify another default time and proceed with the cluster analysis or terminate the session.

You can also use a VAXcluster system analysis when you find a problem on one node and you want to know what was happening on other nodes in the cluster during that time.

13.7 Changing Threshold Values

The Analyzer evaluates performance through tests that compare the values of its performance indicators to values of corresponding metrics in the log file. These performance indicators are called thresholds. For example, in the test "Is page faulting excessive?," the Analyzer compares the value of the threshold High_Fault_Rate to the VAX SPM metric Page Faults. The Analyzer scales the threshold values according to CPU and device type.

You can customize the analysis to your system by changing threshold values. Once you have changed the threshold values, you can save them in a file and use them for subsequent analyses.

Note: **Modification of threshold values can invalidate the results of previous tests. Therefore, it is recommended that if you modify threshold values, you modify them before any tests are run. If you must change threshold values in the middle of a session, set the analysis back to the first test using the RESTART command.**

Use the SHOW THRESHOLD command to display threshold values. You can use the /ALL qualifier to show the values of all thresholds, as in the following example:

```
> SHOW THRESHOLD/ALL
     Threshold Values (* - modified)
        Value    Threshold

      100.00   BASE_HIGH_BUFFERED_IO_RATE
      100.00   BASE_HIGH_FAULT_RATE
        1.00   BASE_HIGH_IMAGE_ACTIVATION_RATE
        1.00   BASE_HIGH_INSWAP_RATE
       1.00    BASE_HIGH_SYS_ACTIVATION_RATE
       50.00   DISK_SATURATION_PCT
       77.00   HIGH_ACTIVATION_PAGING_PCT
       40.00   HIGH_ACTIVE_PROCESS_CPU_PCT
       20.00   HIGH_BORROWING_PCT
      100.00   HIGH_BUFFERED_IO_RATE
       10.00   HIGH_BUSY_WAIT_PCT
        1.00   HIGH_COMPUTE_QUEUE
       15.00   HIGH_DISK_OPERATION_RATE
       35.00   HIGH_DISK_PAGING_PCT
       80.00   HIGH_DISK_READ_PCT
       35.00   HIGH_DISK_SWAPPING_PCT
       80.00   HIGH_DISK_UTIL_PCT
       20.00   HIGH_EXEC_MODE_PCT
      100.00   HIGH_FAULT_RATE
       15.00   HIGH_FCP_DISK_IO_PCT
       50.00   HIGH_FILE_CREATES_PCT
        1.00   HIGH_FREEGOAL_PCT
      150.00   HIGH_FREELIM
       10.00   HIGH_HARD_FAULT_RATE
        1.00   HIGH_IMAGE_ACTIVATION_RATE
        1.00   HIGH_INSWAP_RATE
       15.00   HIGH_INTERRUPT_MODE_PCT
       25.00   HIGH_KERNEL_MODE_PCT
       25.00   HIGH_OUTSWAP_PCT
       15.00   HIGH_PAGE_CACHE_PCT
      800.00   HIGH_PROCESS_SIZE
        0.10   HIGH_SPLIT_IOS_RATE
        5.00   HIGH_SWAPPER_CPU_PCT
        5.00   HIGH_SWAP_WAIT_PCT
       80.00   HIGH_SWPRATE_PCT
        1.00   HIGH_SYS_ACTIVATION_RATE
       60.00   HIGH_TERMINAL_IO_PCT
        0.10   HIGH_WIND_TURNS_RATE
        1.00   LOW_BALSET_CNT
       10.00   LOW_CPU_IDLE_TIME
       70.00   LOW_DIR_CACHE_HIT_PCT
       70.00   LOW_DIR_DATA_CACHE_HIT_PCT
       70.00   LOW_EXTENT_CACHE_HIT_PCT
       70.00   LOW_FILE_HEADER_CACHE_HIT_PCT
       70.00   LOW_FILE_ID_CACHE_HIT_PCT
       25.00   LOW_GLOBAL_VALID_PCT
       10.00   LOW_PRIMARY_CPU_IDLE_PCT
       70.00   LOW_QUOTA_CACHE_HIT_PCT
        0.50   MIN_CACHE_ATTEMPT_RATE
       30.00   MIN_SECONDARY_UTIL_PCT
       25.00   MIN_SYSTEM_TIME_PCT
       25.00   MIN_TASK_TIME_PCT

>
```

You can also select the value of a single threshold for display as in the following command:

```
> SHOW THRESHOLD High_Busy_Wait_Pct
```

```
HIGH_BUSY_WAIT_PCT: 10.00 %
```

Change values of thresholds using the SET THRESHOLD command. It has the following format:

```
SET THRESHOLD [threshold name] [new value]
```

The following command changes the value of the High_Busy_Wait_Pct threshold to 4.0 %:

```
SET THRESHOLD HIGH_BUSY_WAIT_PCT 4.0
```

When you have finished modifying threshold values, you can save them in a file with the SAVE command. In the following example, modified threshold values are saved in an initialization file called SPM$INI_ANALYZE.COM.

```
> SAVE
_Filename: SPM$INI_ANALYZE.COM
Writing changed parameters to SPM$INI_ANALYZE.COM;0...
```

Follow the procedures in Section 13.9 to recall the saved threshold values.

13.8 Modifying Disk and CPU Scaling Factors

The Analyzer evaluates performance through tests that compare the values of its thresholds to values of corresponding metrics in the log file. The default values of these thresholds define the expected performance of a typical system. The values of some of these thresholds are scaled according to the CPU and disk types being analyzed.

CPU scaling is accomplished in this manner. The CPU factor for a VAX-11/780 system is 1.0. CPU factors for all other CPUs are relative to the VAX-11/780. A CPU factor of .75 causes CPU related threshold values to be scaled to .75 of what they are for a VAX-11/780.

Disk scaling is accomplished in this manner. The disk factor is a number representing the maximum operation rate that a disk can maintain before a queue begins to form. The default disk factor for a disk type is a number representing the reciprocal of the average access time for that disk.

The default CPU and disk scaling factors are intentionally conservative. They can be customized, however, for the system being an analyzed. Like threshold values, modified scaling factors can be saved in a file for use in subsequent sessions.

Note: Modification of scaling factor values can invalidate the results of previous tests. Therefore, it is recommended that if you modify scaling factor values, modify them before any tests are run. If you must change the values of scaling factors in the middle of a session, set the analysis back to the first test using the RESTART command.

Use the SHOW DISK_FACTOR command to display the value of disk factors before changing them. The following command displays disk factors for all disks in the system under investigation:

```
> SHOW DISK_FACTOR
```

The SET DISK_FACTOR command has the following format:

```
SET DISK_FACTOR/TYPE=disk_type_name [value]
or
SET DISK_FACTOR disk_name,[disk_name[,...]] [value]
```

Use the SET DISK_FACTOR command to specify disk factor values for disks that are specified by disk type and by name, and to reset a disk factor back to its original value.

The command in the following example changes the disk factor for disks according to disk type:

```
> SET DISK_FACTOR/TYPE=RA81 45.8
  Disk type: RA81 Disk factor 45.8
```

The command below changes the disk factor for a number of disks that are specified by name. Disk names may include the wild card characters * and %.

```
> SET DISK_FACTOR $225$DUA%,DBA6 25
  Disk $255$ DUA6 Disk Factor 25.00 (modified value)
  Disk $255$ DUA6 Disk Factor 25.00 (modified value)
  Disk DBA6 Disk Factor 25 (modified value)
```

Reset a changed disk factor to its original value by specifying the SET DISK_FACTOR command without a value.

Use the SHOW CPU_FACTOR command to show the CPU factor for the current VAXcluster system node before changing it, as seen in the following example:

```
> SHOW CPU_FACTOR
  CPU type: V785 CPU factor: 1.40
```

Use the SHOW CPU_FACTOR command to show the CPU factor for a particular CPU type before changing it. The following is an example:

```
> SHOW CPU_FACTOR/TYPE=8800
  CPU type: 8800 CPU factor: 8.00
```

The SET CPU_FACTOR command has the following format:

```
> SET CPU_FACTOR [/TYPE=cpu_type_name ] [value]
```

Use the SET CPU_FACTOR command to change the CPU factor for the current VAXcluster system node, to change the CPU factor for a particular CPU type, and to reset the CPU factor back to its original value.

To change the CPU factor for the current VAXcluster system node, type the command and a value as shown in the following example:

```
> SET CPU_FACTOR 2
  CPU type: V785 CPU factor: 2.00 (modified value)
```

The CPU factor for the current VAXcluster system node is reset to the original value when new data is read. If no data is read, the changed CPU factor remains in effect until you change or reset it, or until the end of the session.

To modify the CPU factor for a CPU type, you must specify the processor type for that CPU. If you do not know the processor type of the CPU you want to change, use the SHOW DATA_FILE command, which lists processor types for each log file.

The following command modifies the CPU factor for a VAX 8800 system to 7.90.

```
> SET CPU_FACTOR/TYPE=8800 7.90
  CPU type: 8800 CPU factor 7.90 (modified value)
```

An important difference between changing the CPU factor for a VAXcluster system node and changing the CPU factor for a CPU type is that when you change the CPU factor for a CPU type, the value does not get reset when new data is read. It remains in effect until you change or reset it, or until the session ends.

Reset the CPU factor for the current VAXcluster system node to its original value by using the following command:

```
> SET CPU_FACTOR
```

Reset the CPU factor for a CPU type to its original value by typing the SET CPU_FACTOR without specifying a value as shown in the following example:

```
> SET CPU_FACTOR/TYPE=8800
```

13.9 Using an Initialization File to Customize Your Session

You can use an initialization file to customize your analysis session. This file contains comments and SET commands, and is executed during an analysis session.

You can customize your analysis sessions by specifying threshold values for tests and nondefault session characteristics. For example, you can use the SET THRESHOLD command to modify threshold values for your system. You can include this command in the initialization file, or create an initialization file using the SAVE command as described in Section 13.7.

Use the SET CLUSTER_FILES command to define a cluster for analysis. If your cluster is usually active at the same time each day, you can specify that time for analysis using the SET TIME command. If you want a minimum pause between tests when you type the GO command, use the SET PAUSE command to set the value of pause to 0. The following shows an example initialization file:

```
!
! Initialization file for Nodes Purple, Aqua and Pink.
! Pause is set to 0, thresholds are customized for these nodes and
! default time is from 11:45 to 2:45.
!
SET CLUSTER_FILES PURPLE_1988SEP30.DAT,AQUA_1988SEP30.DAT,-
PINK_1988SEP30.DAT
SET PAUSE 0
SET TIME /SINCE=30-SEP-1988:11:45 /BEFORE=30-SEP-1988:14:45
SET THRESHOLD HIGH_FAULT_RATE 150
SET THRESHOLD HIGH_WINDOW_TURNS RATE 1.0
```

There are two ways to execute the initialization file: one way is for an interactive session, the other is for a noninteractive session.

For an interactive analysis session, use the @ command as in the following example:

```
> @Thresholds.com
Reading indirect file DISK$:[USERA]THRESHOLDS.COM
```

For a noninteractive analysis session, use the /COMMAND qualifier to specify the initialization file as in the following example:

```
$ PERFORMANCE ANALYZE=LOG_FILE/COMMAND=THRESHOLDS.COM SPM$COLLECT_TUNE.DAT
```

In the following example, the initialization file SPM$INI_ANALYZE is assumed as the default:

```
$ PERFORMANCE ANALYZE=LOG_FILE/COMMAND SPM$COLLECT_TUNE.DAT
```

In both examples, the command line is placed in a batch command file that is explained in Section 13.3.

14 Reporting Log File Data

This chapter explains how to generate tabular reports and graphs from the data in log files.

It describes the following:

- Invoking the log file reporting command
- Specifying the reporting and graphing period
- Specifying prime and nonprime hours on which to base reports and graphs
- Specifying disks and devices for reports and graphs
- Specifying the reporting interval for reports and graphs
- Using additional reporting features
- Using additional graphing features
- Dumping the data used to generate reports and graphs

14.1 Invoking the Log File Reporting Facility

To generate reports from a log file created with the PERFORMANCE COLLECT=TUNE or PERFORMANCE COLLECT=CAPACITY command, give a command of the following type:

```
PERFORMANCE REPORT=LOG_FILE/qualifiers... logfile-file-spec
```

This command accepts one parameter (the name of the log file) and a number of qualifiers. No special privileges are needed to use this command. The /OUTPUT qualifier can be used to give the file specification for the report; if this is omitted, the report is generated as LOGFILE.RPT in the user default directory. In the case of presentation quality graphs, unique descriptive file names are automatically generated for each graph.

Table 14–1 describes the reporting qualifiers for the REPORT=LOG_FILE command.

Table 14–1 VAX SPM REPORT=LOG_FILE Command Qualifiers

Command Qualifier	Description
/BEFORE	Specifies a range of dates for records selected for reporting. The default is /BEFORE=end of file.
/CLASS=(item[,...])	Specifies the optional classes of statistics to be included in the tabular report or dump file. The default is DEFAULT_STATISTICS.

Table 14–1 (Cont.) VAX SPM REPORT=LOG_FILE Command Qualifiers

Command Qualifier	Description
/DEVICE=([DEVICE[:thresh:scale],...])	Specifies devices for reporting. The default is /DEVICE.
/DISK=([DISK[:thresh:scale],...])	Specifies disks for reporting. The default is /DISK.
/[NO]DUMP=([FILE=name,] [TYPE=[NO]ASCII,] [FORM=format-type])	Dumps report information into a user-specified file. The default is /NODUMP.
/[NO]GRAPH=([graph-item [:thresh:scale], . . .])	Specifies the generation of graphs and selects the items to be graphed. The default is /GRAPH=SUMMARY.
/INTERVAL=[seconds]	Specifies a reporting interval greater than the collection interval of the log file. The default is /INTERVAL=collection interval of log file.
/OUTPUT[=file-spec]	Specifies a report file name. The default is /OUTPUT=LOGFILE.RPT. For presentation quality graphs, a unique descriptive file name is generated for each graph.
/P_HOURS=([range])	Designates hours of the day as prime hours. The default specifies prime hours as 8:00 a.m. to 5:00 p.m.
/[NO]SELECT[=type:percent]	Selectively inhibits graph generation based on user-specified threshold values. The default is /NOSELECT.
/[NO]SHIFT[=shift-type]	Specifies data for reporting based on prime or nonprime hours. The default is /NOSHIFT.
/SINCE[=time]	Specifies a range of dates for records selected for reporting. The default is /SINCE=beginning of file.
/STYLE=type	Specifies working or presentation graphs. The default is /STYLE=TOTAL, which is a working graph in ANSI format.
/[NO]TABULAR=([report-section, . . .])	Specifies the generation of a tabular report and names sections to be included in the report (optional). The default is /TABULAR=FINAL.

14.2 Specifying the Reporting and Graphing Period

Instead of reporting on data for the entire period of time covered by the log file, reports and graphs can be generated for time periods that you select. Use the /BEFORE and /SINCE qualifiers to select a period within a log file for reporting or graphing; both qualifiers accept an argument that is an absolute time, a delta time, or a combination of these two time formats.

The /SINCE qualifier specifies the date/time of the first record in a log file to be processed; if omitted, processing starts with the first record in the file. The /BEFORE qualifier specifies the date/time of the last record in the log file to be processed; if omitted, processing ends with the last record in the file. Omitting both qualifiers causes the entire log file to be processed.

In the following example the reporting period begins with the first record in SPM$COLLECT_TUNE.DAT dated later than 9:45 a.m. on March 17, 1988, and continues until the last record in the file.

```
$ PERFORMANCE REPORT=LOG_FILE/SINCE="17-MAR-1988:9:45"  SPM$COLLECT_TUNE.DAT
```

Times specified with /SINCE and /BEFORE do not have to fall within the range of dates for records in the log file; however, if no records fall within the specified range, the following message is output and the report contains no data:

```
%SPM-F-NODATAFND, No Data Found for Analysis in file
```

14.3 Specifying Prime and Nonprime Hours for Reports and Graphs

The concept of prime time and nonprime time may be used to distinguish periods of peak usage of system resources from periods of lesser usage. For working graphs in ANSI format, prime time is indicated by the small letter "p" under the appropriate interval on the time axis. For tabular statistics, prime and nonprime time data can be generated as separate statistics pages.

Use the PRIME and NONPRIME keywords with the /SHIFT qualifier to select time intervals for reporting based on the concept of prime time.

Using /SHIFT=PRIME generates tabular report statistics pages and graphs that summarize data for prime hours only. Similarly, the /SHIFT=NONPRIME summarizes nonprime hours. Since PRIME and NONPRIME cannot be specified at the same time, there are three possibilities: PRIME time reports, NONPRIME time reports, and reports in which prime and nonprime time is combined (/NOSHIFT).

As shown in Figure 14–1, each 1-hour interval during the day is designated by an integer value in the range 0 through 23. For example, the period 8 a.m. to 9 a.m. is designated by the integer 8; the period 8 a.m. to 12 p.m. is designated by the hours 8–11. By default, prime times are the hours 8–16, or 8 a.m. to 5 p.m. Times outside these hours fall into the nonprime time category.

Figure 14–1 Integer Values of Hours During the Day

```
        0     1     2         7     8     9    10    11    12   13   23   24
TIME |-----|-----|----|...|-----|-----|-----|-----|-----|-----|...|-----|
HOUR   0     1     2         7     8     9    10    11    12         23
```

Each 1-hour interval during the day is designated by an integer value in the range 0 through 23. For example, the period 8 a.m. to 9 a.m. is designated by the integer 8; the period 8 a.m. to 12 p.m. is designated by the range 8–11. By default, prime time is the range 8–16, or 8 a.m. to 5 p.m. Hours outside this range fall into the nonprime time category.

The definition of prime time can be changed using the /P_HOURS qualifier; this qualifier accepts integer values or ranges that override the default definition of prime time.

The /P_HOURS qualifier has the following format:

```
/P_HOURS=([range...])
```

The /P_HOURS qualifier specifies one or more ranges of hours to be prime time. A range of hours is specified as two integer values separated by a hyphen (-). Multiple hours and hour ranges are separated by commas.

The following example specifies prime hours of 0800 to 1100 (8 a.m. to 11 a.m.) and 1300 to 1800 (1 p.m. to 6 p.m.).

```
/P_HOURS=(8-10,13-17)
```

If this qualifier is omitted from the command line, the default of /P_HOURS=(8–16) (prime time of 8 a.m. to 5 p.m.) is assumed.

14.4 Specifying Disks and Devices for Reports and Graphs

Use the /DEVICE and /DISK qualifiers to select those devices and disks that are to be included in the graph and tabular sections of the report or the dump file. Device and disk names must be physical names, but it is not necessary to include a node prefix if one is present in the log file. Physical names can be truncated to specify a group of devices or disks. When a truncated disk or device name is supplied, one of three actions is taken:

- If the disk or device name is truncated (for example, DB or XE), then reporting is done for all disks or devices beginning with what was entered (in this example, DB or XE).
- If the controller designation is omitted (for example, DB2 or XE1), then reporting is done for all disks or devices on all controllers with the specified unit number (for example, DBA2 and DBB2).
- If the unit number is omitted (for example, DBA or XEA), then reporting is done for all disks or devices on the specified controller (for example, DBA1 and DBA2).

By default, all devices and disks will be graphed if an applicable graph item is specified using /GRAPH. The keywords beginning with D_ (for example, D_UTILIZATION) are the graph items that apply to disks, and cause disk data such as storage allocation, I/O rate, response time, and utilization to be plotted. The D_ keywords can be accompanied by a threshold value as described in Section 14.7.4.1.

To specify individual devices and disks, use the /DEVICE and /DISK qualifiers in addition to /GRAPH. For example, specifying /GRAPH=DEVICE/DEVICE=TTA6 causes the device I/O rate for device TTA6 to be plotted. The DEVICE keyword can also be accompanied by a threshold value as described in Section 14.7.4.1.

Although all devices and disks will be graphed by default, Section 14.7.4 describes how to inhibit the generation of graphs for devices and disks based upon thresholds you define.

The following command generates tabular, graphic disk utilization, and rate statistics for all the DU disks (for example, $255DUA0, DUA0, and DUA1).

```
$ PERFORMANCE REPORT=LOG_FILE/DISK=DU/GRAPH=(D_UTILIZATION,D_RATE)/TABULAR -
  SPM$COLLECT_TUNE.DAT
```

14.5 Specifying an Interval Longer Than the Log File Collection Interval

You can report or graph at intervals that are longer than the log file collection interval by specifying a value for the /INTERVAL qualifier. When you specify a reporting interval longer than the collection interval of the log file, data for intervals in the log file is combined into longer intervals for reporting.

If the interval you specify is not an even multiple of the collection interval, it is adjusted up to the next whole multiple of the log file's collection interval.

For example, if a reporting interval of 600 seconds is specified rather than the log file collection interval, the reporting interval is adjusted up to the next whole multiple of the log file's collection interval. If the log file collection interval was 300 seconds, interval statistics pages will report statistics for each 600-second interval (two log file collection intervals). If the collection interval is 250 seconds, however, statistics are reported for each 750 seconds (three collection intervals).

If /INTERVAL is omitted from the command line or no value is specified, the default report or graph interval is the collection interval of the log file.

The value of the /INTERVAL qualifier and the length of the reporting period determine the number of intervals reported when /TABULAR=INTERVAL is specified as described in Section 14.6.2. This equation illustrates the relationship of the value of /INTERVAL and the reporting unit.

```
REPORTING PERIOD
---------------- =  Number of interval reports
value of /INTERVAL  when /TABULAR=INTERVAL
```

For example, if the reporting period is one day (24 hours) and the value specified for the /INTERVAL qualifier is 900 seconds (15 minutes), 96 interval tabular reports are produced when /TABULAR=INTERVAL is specified.

In the following example, interval statistics are reported for log file SPM$COLLECT_TUNE.DAT with a report interval of 900. Assuming the log file collection interval of 300 seconds, an interval is reported for every three log file collection intervals. This command generates one page for each 900-second (15-minute) interval within the reporting period of one hour.

```
$ PERFORMANCE REPORT=LOG_FILE/TABULAR=INTERVAL -
  /INTERVAL=900/BEFORE="30-AUG-1988:11:00" -
  /SINCE="30-AUG-1988:10:00" SPM$COLLECT_TUNE.DAT
```

The following command generates graphs with a report interval of 1000 seconds; assuming a log file collection interval of 300 seconds, there is one plotted value over four samples (1200 seconds = new time unit).

```
$ PERFORMANCE REPORT=LOG_FILE/INTERVAL=1000/NOTABULAR -
  /GRAPH=(CPU,DIRECT_IO) SPM$COLLECT_TUNE.DAT
```

In the case of graphs, the reporting unit is the entire graph, and each interval is represented by a column. The number of columns is determined by how many intervals there are in the reporting period.

In the command below, the reporting period is one hour and the interval is for 900 seconds (15 minutes). The graph represents one hour; each column on the graph represents one 15-minute interval.

```
$ PERFORMANCE REPORT=LOG_FILE/NOTABULAR/GRAPH/INTERVAL=900 -
        /BEFORE="30-AUG-1988:11:00" -
        /SINCE="30-AUG-1988:10:00" SPM$COLLECT_TUNE.DAT
```

14.6 Additional Reporting Features

Use the /TABULAR qualifier with the PERFORMANCE REPORT=HISTORY or the PERFORMANCE REPORT=LOG_FILE command to generate tabular reports.

In addition to the reporting capabilities described in previous sections, this section describes the following:

- Reporting on various classes of performance metrics
- Producing interval reports to report performance metrics for each interval and producing final reports for entire reporting periods
- Producing interval and final reports for prime and nonprime hours

Read the *VAX SPM Reference Manual* for a full description of tabular reports and their contents.

14.6.1 Specifying the Metrics for a Tabular Report

Use the /CLASS qualifier with the /TABULAR qualifier to control the classes of statistics that appear in the tabular reports. The /CLASS qualifier has the following format:

```
/CLASS=(item,[...])
```

Using the /CLASS qualifier, all default and optional data present in the log file can be reported. Keywords for the /CLASS qualifier are described in Table 14–2.

If /CLASS is omitted from the command line, only default statistics (CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE) are reported.

Table 14–2 Keywords for the /CLASS Qualifier

Keyword	Meaning
ALL	All optional statistics
CPU	CPU statistics
DEFAULT_STATISTICS	CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE statistics
DEVICE	Device rate/second statistics for devices specified by /DEVICE
DISK	Disk statistics for disks specified by /DISK
FILE_PRIMITIVES	File system primitive counts
GLOBAL_SECTIONS	Global section information
HARDWARE_ CONFIGURATION	Hardware configuration information
INSTALLED_IMAGES	Installed images information
IO	I/O statistics
LOCK	Lock statistics
MEMORY	Memory statistics
PAGE_FAULT	Page fault statistics
PROCESS[=[NO]EXTENDED, [NO]IMAGE,ALL]	Process statistics
SCHEDULER	Scheduler statistics
SYSGEN_PARAMETERS	An alphabetical list of system parameters and their values
SYSTEM_COMMUNICATION	System communication services data
XQP_CACHE	XQP caching statistics

All the above keywords are available in negated form. For example, /CLASS =(ALL,NOLOCK) will include all statistics, default and optional classes, except lock statistics. Note that the DEFAULT_STATISTICS class of statistics is always reported unless specifically negated. Reporting of default statistics can be disabled wholly or partially by negating and/or including specific keywords. For example, to report only disk and CPU data, /CLASS=(NODEFAULT_STATISTICS, DISK,CPU) can be specified.

In the example below, default statistics and lock statistics for the log file SPM$COLLECT_TUNE are reported. Note that the default statistics CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE are always reported unless specifically negated.

```
$ PERFORMANCE REPORT=LOG_FILE/CLASS=LOCK/TABULAR=FINAL -
 /NOGRAPH  SPM$COLLECT_TUNE.DAT
```

In the following example, default statistics and all optional statistics (except lock statistics) for SPM$COLLECT_TUNE.DAT are reported:

```
$ PERFORMANCE REPORT=LOG_FILE/CLASS=(ALL,NOLOCK) -
 /NOGRAPH SPM$COLLECT_TUNE.DAT
```

Note that in the above examples, the /NOGRAPH qualifier suppresses the generation of graphs.

14.6.2 Specifying the Type of Tabular Report

The /TABULAR qualifier controls not only the presence of tabular data in a report, but also the intervals for reporting data. With regard to time, there are two types of reporting (final statistics and interval statistics) as follows:

1. Final statistics are for the entire reporting period. Each item of tabular data appears just once in the report. Figure 14–2 shows the first page of a Final Statistics report.
2. Interval statistics are for a specific interval. Each item of tabular data appears as many times as there are intervals in the reporting period; therefore, Interval Statistics reports may produce a significant amount of output.

Keywords used with /TABULAR are shown in Table 14–3.

Table 14–3 Keywords for the /TABULAR Qualifier Common to Both Commands

Keyword	Meaning
ALL	All keyword options shown below
FINAL	Final statistics page summarizing data across entire reporting period
INTERVAL	Statistics page for each reporting interval

If no keyword is specified with the /TABULAR qualifier, /TABULAR=FINAL is assumed.

All keywords are available in negated form.

The FINAL keyword causes data to be reported over the entire reporting period. For example, if the collection interval in a log file is 300 seconds (5 minutes) and you specify final statistics for a reporting period of one hour, one final statistics page would be generated. The following is an example of the command:

```
$ PERFORMANCE REPORT=LOG_FILE/SINCE="30-AUG-1988:10:00" -
 /BEFORE="30-AUG-1988:11:00"/TABULAR=FINAL/NOGRAPH SPM$COLLECT_TUNE.DAT
```

The INTERVAL keyword with the /TABULAR qualifier causes tabular data to be reported for each interval in a reporting period, rather than over the entire reporting period itself. For each reporting interval, there is a separate statistics page.

For example, substituting INTERVAL for FINAL in the above command for the same log file would produce 12 interval statistics pages, one for each collection interval in the log file. This is because the default interval is 5 minutes (/INTERVAL=300 seconds is assumed). The following command is an example:

```
$ PERFORMANCE REPORT=LOG_FILE/SINCE="30-AUG-1988:10:00" -
 /BEFORE="30-AUG-1988:11:00"/TABULAR=INTERVAL/NOGRAPH SPM$COLLECT_TUNE.DAT
```

Note in both commands that the /NOGRAPH qualifier suppresses the generation of graphs.

You can change the size of the interval for which statistics are reported by stating a value for the /INTERVAL qualifier, which is described in Section 14.5.

14.6.3 Specifying Final and Interval Reports for Prime and Nonprime Hours

The /SHIFT qualifier may be specified in a number of ways to select data for reporting. In the following examples:

/TABULAR=FINAL/NOSHIFT generates final statistics where the data from prime and nonprime time is reported.

/TABULAR=FINAL/SHIFT=PRIME produces final prime statistics.

The substitution of the NONPRIME keyword for PRIME in the above example generates nonprime statistics.

In the example below, prime time is the period from 8 a.m. to 5 p.m. (the default). The report will contain final statistics for the entire reporting period (the period of time covered by records in SPM$COLLECT_TUNE.DAT) for prime time only. Only CPU statistics are specified.

```
$ PERFORMANCE REPORT=LOG_FILE/CLASS=(NODEFAULT_STATISTICS,CPU) -
/TABULAR=FINAL/SHIFT=PRIME SPM$COLLECT_TUNE.DAT
```

In the example below, prime time is defined to be the period from 8 a.m. to 12 p.m., and from 1 p.m. to 5 p.m. The report will contain a final statistics page (default statistics only) summarized for these hours only, across the period of time covered by records in SPM$COLLECT_TUNE.DAT.

```
$ PERFORMANCE REPORT=LOG_FILE/TABULAR=FINAL/SHIFT=PRIME -
  /P_HOURS=(8-11,13-16) SPM$COLLECT_TUNE.DAT
```

Figure 14–2 Final Statistics Report Example

```
Report Date:  2-MAR-1989 14:09                       GREEN                                   VAX SPM V3.3         Page     2

************************                            FINAL Statistics                                     ***********************
************************    Data Analyzed: from  14-NOV-1988 07:55:56.20  to  14-NOV-1988 17:05:00.04    ***********************

+--- AVE Process-Memory Counts ---+------- Memory Utilization -------+- AVE Mem/CPU -+------------- Swapper Counts -------------+
!                                 !                                  !    Queues     !                                          !
!  Proc    Balset   Free   Modify !  Total   Paged   User   Modify   !               !                    Header  Header  Swaper !
!  Count   Count    Pages   Pages ! MEMutl  MEMutl  MEMutl  MEMutl   !  Mem     CPU  !  InSWP   OutSWP  InSWP   OutSWP  CPU %  !
!  ------  ------  -------  ------ ! ------  ------  ------  ------   ! -----  ----- !  ------  ------  ------  ------  ------ !
!    71      68    50144     888  !  49.0%   40.9%   41.7%    1.0%   !    0       0 !     0       0       0       0     0.4%  !
+---------------------------------+----------------------------------+---------------+------------------------------------------+

+-------------------------- CPU Statistics -------------------------+------------------+
!  CPU   Total   Busy  Inter                                       !                  !
!   ID    Idle   Wait  Stack   Kernel    Exec   Super   User Compat !  System   Task   !
!  ---  ------  ------ -----   ------  ------  ------  ------  ------ !  ------  ------  !
!    7   66.3%    3.2%  4.2%    13.0%    2.8%    0.4%   13.3%   0.0% !   20.0%   13.7%  !
!    0   88.6%    0.0%  0.0%     0.1%    0.6%    0.2%   10.5%   0.0% !    0.8%   10.7%  !
+-------------------------------------------------------------------+------------------+

+-------- Lost CPU --------+----------- CPU and I/O Overlap ----------+
!                   Page   !                                         !
!   Page    Swap   or Swp  !  CPU+IO   CPU     I/O    Multi   CPU+IO !
!   Wait    Wait    Wait   !   Idle    Only    Only    I/O     Busy  !
!  ------  ------  ------  !  ------  ------  ------  ------  ------ !
!   5.1%    2.1%    6.9%   !  32.6%   20.2%   29.5%   12.5%   17.7%  !
+--------------------------+-----------------------------------------+

+------------------------------------- Paging Rates (per second) -------------------------------------+     +----------------+
!                                                                                                     !     !                !
!  Page   System   Pages    Read   Pages   Write   Free   Modify    Bad   Dzero  Gvalid  Trans WritIN !     !  Hard    Soft  !
! Faults  Faults   Read    I/Os   Writen   I/Os    List    List    List  Faults  Faults  Faults  Prog  !     ! Faults  Faults !
! ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------ !     ! ------  ------ !
!  180.9     1.0    32.2     2.4    38.4     0.4    48.3    30.8     0.0    75.7    22.9     0.0    0.7 !     !   1.3%   98.7% !
+-----------------------------------------------------------------------------------------------------+     +----------------+

+--------- I/O Rates (per second) ---------+              +------ File I/O Rates (per second) -----+         +----------+
!                                          !              !                                        !         !   AVE    !
!  Direct  Buffrd  Lognam  Mailbx  Mailbx  !              ! Window  Window   Split   Erase   File  !         !   Open   !
!   I/Os    I/Os    Trans   Reads  Writes  !              !  Hits    Turns    I/Os    I/Os   Opens !         !   Files  !
!  ------  ------  ------  ------  ------  !              ! ------  ------  ------  ------  ------ !         !  ------  !
!    17.3    32.2    23.3     1.6     1.5  !              !   19.7     1.0     1.5     0.1     1.7 !         !    238   !
+------------------------------------------+              +----------------------------------------+         +----------+

+---------- File Cache Attempt Rate (per second) ----------+          +---------------- File cache Effectivness ----------------+
!                                                          !          !                                                        !
!   Dir     Dir             File    File              Bit  !          !   Dir     Dir              File    File            Bit  !
!   FCB     Data    Quota    Id      Hdr    Extent    Map  !          !   FCB     Data    Quota     Id      Hdr    Extent   Map  !
!  ------  ------  ------  ------  ------  ------  ------  !          !  ------  ------  ------  ------  ------  ------  ------ !
!     5.3     9.5     0.0     0.2     3.6     0.9     0.6  !          !  99.1%   93.5%    0.0%   99.3%   79.0%   96.8%    2.2% !
+----------------------------------------------------------+          +--------------------------------------------------------+

+--------------------------------------------- Disk Statistics ---------------------------------------------+
!                                   Work                                                 Serv    Resp                        !
!                                   Avail  Paging  Swping  Contlr   Rate    Read   Remote  Time    Time   Queue   Space      !
!                                    %       %       %       %      (/s)     %     I/O%    (ms)    (ms)   Length  Used %     !
!                                  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------    !
! $255$DUA0                         11.2    51.2     1.2    52.8     4.5  ......  ......     25      28     0.1    92.2      !
! $255$DUA1                          1.0     0.2     0.0     2.2     0.4  ......  ......     25      25     0.0    77.1      !
```

Note that the INTERVAL keyword for the /TABULAR qualifier could be used with /SHIFT=PRIME to produce interval statistics for prime hours. Likewise, using /TABULAR=INTERVAL and the /NOSHIFT qualifiers together could produce interval statistics in which data from prime and nonprime time is averaged together. Since a log file can contain a lot of intervals, it is more efficient to specify a period of interest using the /BEFORE and /SINCE qualifiers as described in Section 14.2.

14.7 Additional Graphing Features

Use the /GRAPH qualifier with the PERFORMANCE REPORT=LOG_FILE command to graph performance data.

Read the *VAX SPM Reference Manual* for a full description of graphs and their contents.

In addition to using the features described in Section 14.1 through Section 14.5, you can also:

- Generate graphs for a variety of system metrics.
- Generate working graphs in ANSI format or presentation graphs in ReGIS or sixel format.
- Specify the time intervals to graph.
- Specify threshold values for graph statistics.
- Selectively inhibit graph generation based upon threshold values you define.
- Specify a maximum graph scaling value to override the autoscale feature.

14.7.1 Specifying the System Metrics to Graph

The /GRAPH qualifier specifies that a hard copy graph is desired, and names the types of items to be graphed. If /GRAPH is omitted, or specified without a graph item, a graph showing CPU utilization is produced. If /GRAPH is specified, any of the graph items shown in Table 14–4 can be used as graph item keywords. If you specify the keyword ALL, you can exclude selected graphs by also specifying their keywords prefixed by NO.

Table 14–4 Keywords for the /GRAPH Qualifier

Keyword	Meaning
ALL	All optional graphs
BALANCE_SET	Average number of processes in balance set
BUFFERED_IO	Average number of buffered I/O requests
CPU	Percentage of time the CPU was busy
DEVICE	Device I/O rate per second

Table 14–4 (Cont.) Keywords for the /GRAPH Qualifier

Keyword	Meaning
DIRECT_IO	Average number of direct I/O requests
D_ALLOCATION	Percentage of disk volume storage allocated
D_RATE	Average number of I/O requests per second
D_RESPONSE	Average disk response time in milliseconds
D_UTILIZATION	Percentage of time the disk was busy
FILE_OPENS	Average number of file open requests
LOGICAL_NAME	Average number of logical name translation requests
LOST_TIME	Percentage of time CPU waited for paging and/or swapping I/O to complete
MAILBOX_READS	Average number of mailbox read I/O requests
MEMORY	Percentage of memory utilized
OPEN_FILES	Average number of files open at the same time
PAGE_FAULTS	Average page fault rate
PROCESS_COUNT	Average number of processes in the system
Q_CPU	Average length of the CPU queue
Q_MEMORY	Average length of the memory queue
SUMMARY	Percentage graph of the CPU+IO overlap
SWAP_COUNT	Average number of swapped processes
WINDOW_HIT	Percent ratio of window hits to window turns

The command below generates a working style graph in ANSI format showing the average number of processes in the balance set. The time period for data reporting as specified by the /SINCE and /BEFORE qualifiers is one day; therefore, the report will contain one graph. Note that if the number of intervals in the reporting period exceeds 120, the graph is printed onto additional pages. The /NOTABULAR qualifier suppresses generation of tabular reports.

```
$ PERFORMANCE REPORT=LOG_FILE/SINCE="02-JUL-1986:0:0:0.0" -
  /BEFORE="03-JUL-1986:0:0:0.0"/GRAPH=(BALANCE_SET) -
  /NOTABULAR SPM$COLLECT_CAPACITY.DAT
```

The command in the following example generates working graphs in ANSI format showing CPU utilization and Direct I/O requests plotted against time of day; there is one plotted value per log file collection interval (default time unit).

```
$ PERFORMANCE REPORT=LOG_FILE/GRAPH=(CPU,DIRECT_IO) -
 /NOTABULAR  SPM$COLLECT_TUNE.DAT
```

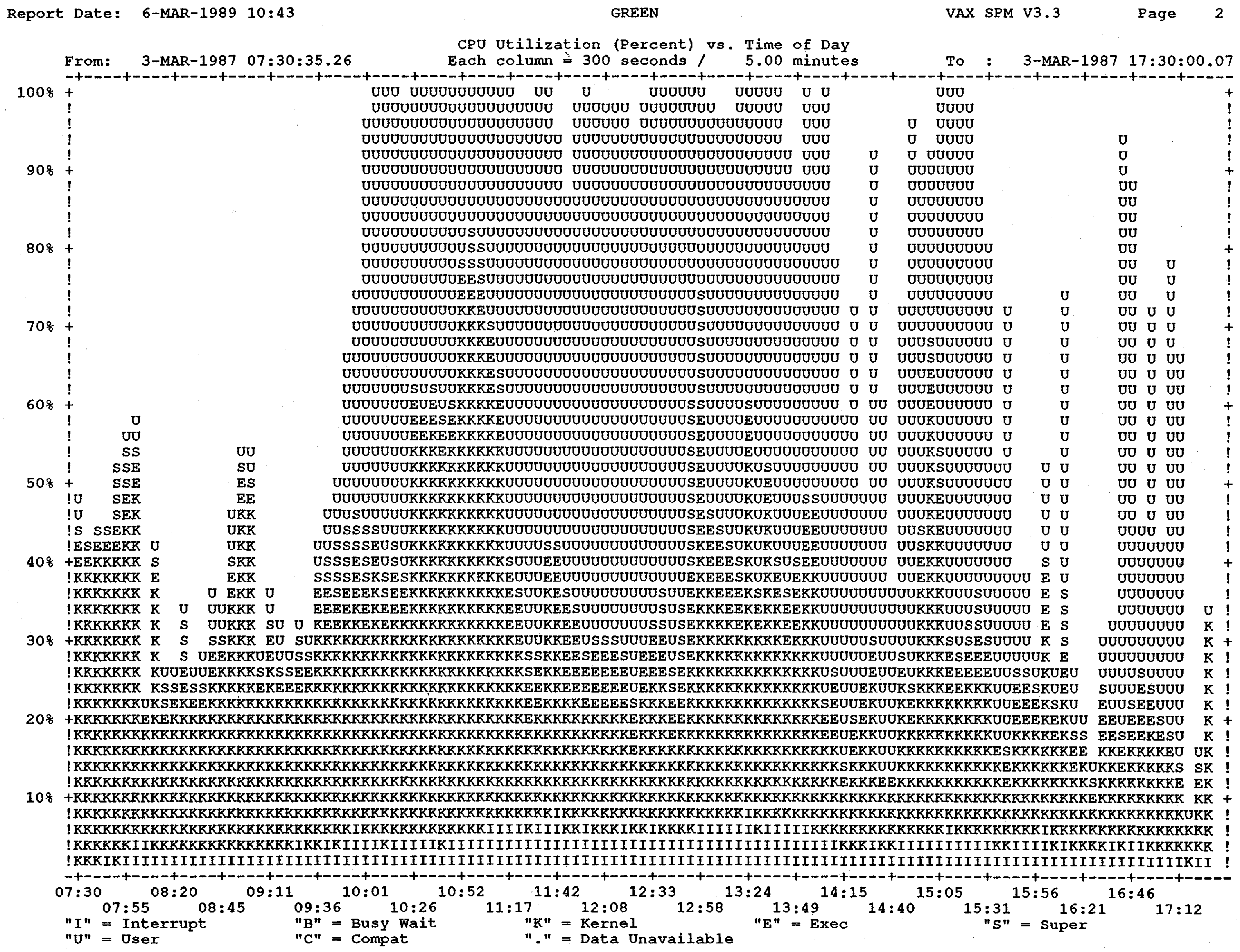
Several different types of graphs are possible, depending on the data to be plotted. System metrics values are plotted along the y axis, either as percentages (0%–100%) or using an autoscale that adjusts itself to the actual values present. Section 14.7.5 describes the autoscaling feature and how to override it. Plotting characters can be simple or compound. With simple plotting characteristics, a column of asterisks is used. With compound plotting characteristics, y axis values are shown as comprised of several different parts using different characters to produce a "column" value. For example, CPU utilization may show a graph column made up of the characters U (user), K (kernel), and I (interrupt stack). Time intervals are plotted along the x axis.

Figures 14–3 and 14–4 show examples of graphs generated for CPU utilization (% y axis, compound plotting) and Direct I/O statistics (autoscale y axis, simple plotting).

14.7.2 Specifying the Style of Graphs

Use the /STYLE qualifier with one of the four keywords shown in Table 14–5 to specify one of four different styles of graph.

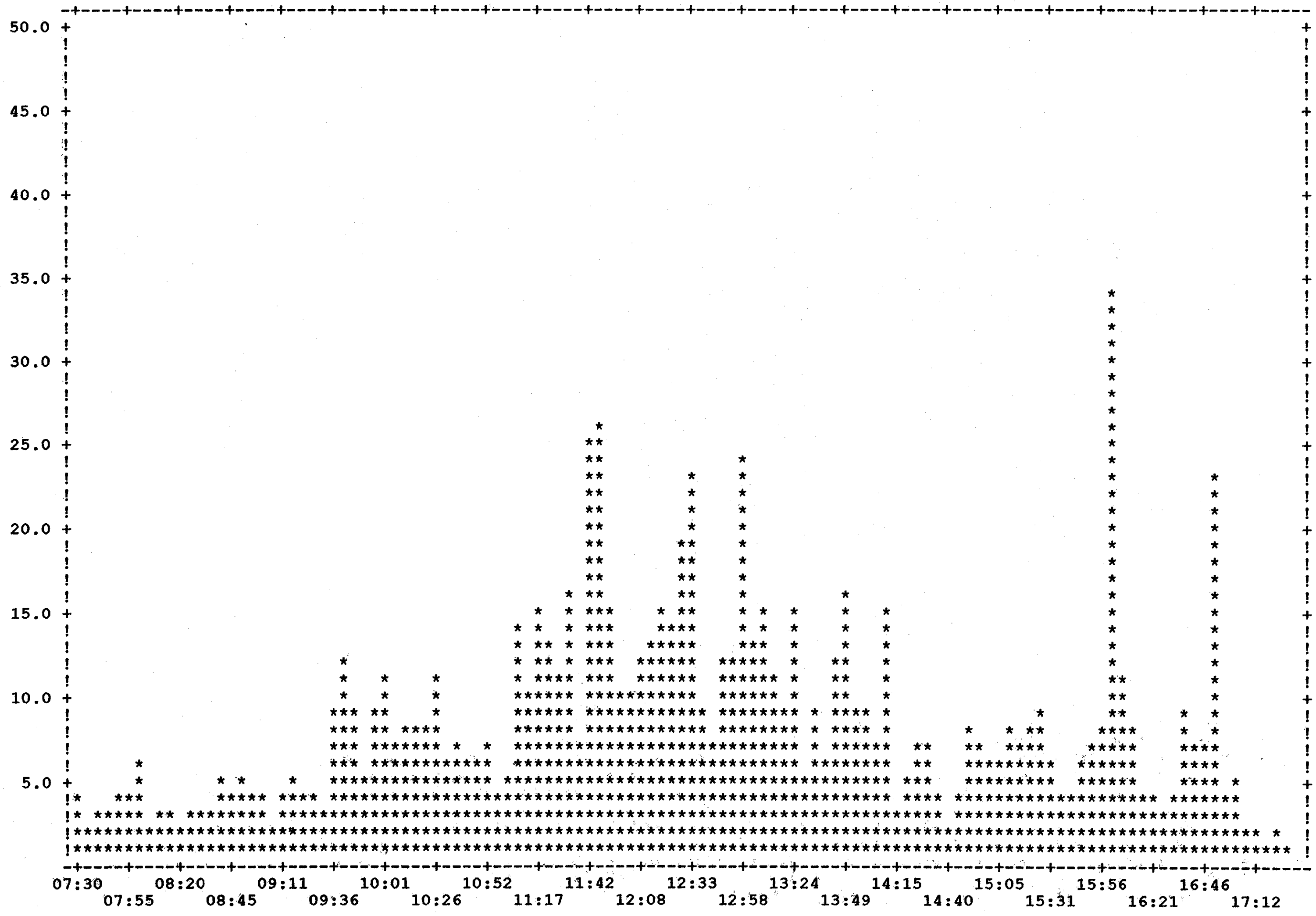
Figure 14–3 CPU Utilization Report Graph Example



```
Report Date:  6-MAR-1989 10:43                                 GREEN                             VAX SPM V3.3        Page    2
                                               CPU Utilization (Percent) vs. Time of Day
      From:   3-MAR-1987 07:30:35.26          Each column = 300 seconds /    5.00 minutes         To   :  3-MAR-1987 17:30:00.07
      -+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-----
 100% +                               UUU UUUUUUUUUUU  UU   U      UUUUUU   UUUUU  U U           UUU                           +
      !                               UUUUUUUUUUUUUUUUUUU  UUUUUU UUUUUUUU  UUUUU  UUU           UUUU                          !
      !                              UUUUUUUUUUUUUUUUUUUU  UUUUUU UUUUUUUUUUUUUUU  UUU        U  UUUU                          !
      !                              UUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUU  UUU        U  UUUU               U          !
      !                              UUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUU UUU    U   U UUUUU               U          !
  90% +                              UUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUU UUU    U   UUUUUUU               U          +
      !                              UUUUUUUUUUUUUUUUUUUUU UUUUUUUUUUUUUUUUUUUUUUUUUUU    U   UUUUUUU               UU         !
      !                              UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU    U   UUUUUUUU              UU         !
      !                              UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU    U   UUUUUUUU              UU         !
      !                              UUUUUUUUUUUSUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU    U   UUUUUUUU              UU         !
  80% +                              UUUUUUUUUUSSUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU    U   UUUUUUUUU             UU         +
      !                              UUUUUUUUUUSSSUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU   U   UUUUUUUUU             UU   U     !
      !                              UUUUUUUUUUEESUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU   U   UUUUUUUUU             UU   U     !
      !                             UUUUUUUUUUUEEEUUUUUUUUUUUUUUUUUUUUUUSUUUUUUUUUUUUUU   U   UUUUUUUUU       U     UU   U     !
      !                             UUUUUUUUUUUKKEUUUUUUUUUUUUUUUUUUUUUUSUUUUUUUUUUUUUU U U  UUUUUUUUUU U     U     UU U U     !
  70% +                             UUUUUUUUUUUKKKSUUUUUUUUUUUUUUUUUUUUUSUUUUUUUUUUUUUU U U  UUUUUUUUUU U     U     UU U U     +
      !                             UUUUUUUUUUUKKKEUUUUUUUUUUUUUUUUUUUUUSUUUUUUUUUUUUUU U U  UUUSUUUUUU U     U     UU U U     !
      !                            UUUUUUUUUUUUKKKEUUUUUUUUUUUUUUUUUUUUUSUUUUUUUUUUUUUU U U  UUUSUUUUUU U     U     UU U UU    !
      !                            UUUUUUUUUUUUKKKESUUUUUUUUUUUUUUUUUUUUSUUUUUUUUUUUUUU U U  UUUEUUUUUU U     U     UU U UU    !
      !                            UUUUUUUSUSUUKKKESUUUUUUUUUUUUUUUUUUUUSUUUUUUUUUUUUUU U U  UUUEUUUUUU U     U     UU U UU    !
  60% +                            UUUUUUUEUEUSKKKKEUUUUUUUUUUUUUUUUUUUSSUUUUSUUUUUUUUU U UU UUUEUUUUUU U     U     UU U UU    +
      !      U                     UUUUUUUEEESEKKKKEUUUUUUUUUUUUUUUUUUUSEUUUUEUUUUUUUUUUU UU UUUKUUUUUU U     U     UU U UU    !
      !     UU                     UUUUUUUEEKEEKKKKEUUUUUUUUUUUUUUUUUUUSEUUUUEUUUUUUUUUUU UU UUUKUUUUUU U     U     UU U UU    !
      !     SS          UU         UUUUUUUKKKEKKKKKKUUUUUUUUUUUUUUUUUUUSEUUUUEUUUUUUUUUUU UU UUUKSUUUUU U     U     UU U UU    !
      !    SSE          SU         UUUUUUUKKKKKKKKKKUUUUUUUUUUUUUUUUUUUSEUUUUKUSUUUUUUUUU UU UUUKSUUUUUUU   U U     UU U UU    !
  50% +    SSE          ES        UUUUUUUUKKKKKKKKKKUUUUUUUUUUUUUUUUUUUSEUUUUKUEUUUUUUUUU UU UUUKSUUUUUUU   U U     UU U UU    +
      !U   SEK          EE        UUUUUUUUKKKKKKKKKKUUUUUUUUUUUUUUUUUUUSEUUUUKUEUUUSSUUUUUUU UUUKEUUUUUUU   U U     UU U UU    !
      !U   SEK         UKK       UUUSUUUUUKKKKKKKKKKUUUUUUUUUUUUUUUUUUUSESUUUKUKUUUEEUUUUUUU UUUKEUUUUUUU   U U     UU U UU    !
      !S SSEKK         UKK       UUSSSSUUUKKKKKKKKKKUUUUUUUUUUUUUUUUUUUSEESUUKUKUUUEEUUUUUUU UUSKEUUUUUUU   U U     UUUU UU    !
      !ESEEEKK U       UKK      UUSSSSEUSUKKKKKKKKKKUUUUSSUUUUUUUUUUUUUSKEESUKUKUUUEEUUUUUUU UUSKKUUUUUUU   U U     UUUUUUU    !
  40% +EEKKKKK S       SKK      USSSESEUSUKKKKKKKKKKSUUUEEUUUUUUUUUUUUUSKEEESKUKSUSEEUUUUUUU UUEKKUUUUUUU   S U     UUUUUUU    +
      !KKKKKKK E       EKK      SSSSESKSESKKKKKKKKKKEUUUEEUUUUUUUUUUUUUEKEEESKUKEUEKKUUUUUUU UUEKKUUUUUUUUU E U     UUUUUUU    !
      !KKKKKKK K     U EKK U    EESEEEKSEEKKKKKKKKKKESUUKESUUUUUUUUUSUUEKKEEEKSKESEKKUUUUUUUUUUKKKUUUSUUUUU E S     UUUUUUU    !
      !KKKKKKK K  U  UUKKK U    EEEEKEKEEEKKKKKKKKKKEEUUKEESUUUUUUUUSUSEKKKEEKEKKEEKKUUUUUUUUUUKKKUUUSUUUUU E S     UUUUUUU  U !
      !KKKKKKK K  S  UUKKK SU U KEEKKEKEKKKKKKKKKKKKEEUUKKEEUUUUUUUUSSUSEKKKEKEKKEEKKUUUUUUUUUUKKKUUSSUUUUU E S    UUUUUUUU  K !
  30% +KKKKKKK K  S  SSKKK EU SUKKKKKKKKKKKKKKKKKKKKKEUUKKEEUSSSUUUEEUSEKKKKKKKKKEKKKUUUUUSUUUUKKKSUSESUUUU K S   UUUUUUUUU  K +
      !KKKKKKK K  S UEEKKKUEUUSSKKKKKKKKKKKKKKKKKKKKKSSKKEESEEESUEEEUSEKKKKKKKKKKKKKKUUUUUEUUSUKKKESEEEUUUUUK E   UUUUUUUUU  K !
      !KKKKKKK KUUEUUEKKKSKSSEEKKKKKKKKKKKKKKKKKKKKKKSEKKEEEEEEEEUEEESEKKKKKKKKKKKKKUSUUUEUUEUKKKEEEEEEUUSSUKUEU  UUUUSUUUU  K !
      !KKKKKKK KSSESSKKKKKEKEEEKKKKKKKKKKKKKKKKKKKKKKEEKKEEEEEEEEUEKKSEKKKKKKKKKKKKKKUEUUEKUUKSKKKEEKKKUUEESKUEU  SUUUESUUU  K !
      !KKKKKKKUKSEKEEKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKEEKKKKEEEEESKKKEEKKKKKKKKKKKKKKSEUUEKUUKEKKKKKKKKUUEEEKSKU  EUUSEEUUU  K !
  20% +KKKKKKKEKEKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKEKKKKKKKKKEKKKEEKKKKKKKKKKKKKKEEUSEKUUKEKKKKKKKKUUEEEKEKUU EEUEEESUU  K +
      !KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKEKKKEKKKKKKKKKKKKKKKEEUEKKUUKKKKKKKKKKUUKKKKEKSS EESEEKESU  K !
      !KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKEKKKKKKKKKKKKKKKKKKKKKUEKKUUKKKKKKKKKKESKKKKKKEE KKEKKKKEU UK !
      !KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKSKKKUUKKKKKKKKKKKEKKKKKKKEKUKKEKKKKKS SK !
      !KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKEKKKEEKKKKKKKKKKEKKKKKKKKSKKKKKKKKE EK !
  10% +KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKEKKKKKKKKK KK +
      !KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKIKKKKKKKKKKKKKKKKKKKKIKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKUKK !
      !KKKKKKKKKKKKKKKKKKKKKKKKKKKKKIKKKKKKKKKKKKKIIIIKIIIKKIKKKIKKKIKKKKIIIIIIKIIIIIKKKKKKKKKKKKKIKKKKKKKKKIKKKKKKKKKKKKKKKKK !
      !KKKKKKIIKKKKKKKKKKKKKKKIKKIKIIIIKIIIIIKIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIKKKIKKIIIIIIIIIIIIIKKIIIIKIKKKKIKIIKKKKKKK !
      !KKKIKIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIKII !
      -+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-----
     07:30     08:20     09:11     10:01     10:52     11:42     12:33     13:24     14:15     15:05     15:56     16:46
          07:55     08:45     09:36     10:26     11:17     12:08     12:58     13:49     14:40     15:31     16:21     17:12
      "I" = Interrupt         "B" = Busy Wait         "K" = Kernel            "E" = Exec              "S" = Super
      "U" = User              "C" = Compat            "." = Data Unavailable
```

Figure 14–4 Direct I/O Statistics Report Graph Example

```
Report Date:  6-MAR-1989 10:43                      GREEN                          VAX SPM V3.3         Page    3
                                           Direct I/Os (Rate/Second) vs. Time of Day
          From:   3-MAR-1987 07:30:35.26          Each column = 300 seconds /    5.00 minutes        To  :   3-MAR-1987 17:30:00.07
```



```
"*" = Direct I/O          "." = Data Unavailable
```

Table 14–5 Keywords for the /STYLE Qualifier

Keyword	Meaning
TOTAL	Working style graph in ANSI format showing the total value for each graph interval
RANGE	Working style graph in ANSI format showing the average values for each graph interval
SIXEL	Presentation graphs produced in sixel output format
REGIS	Presentation graphs produced in ReGIS output format

14.7.2.1 Specifying Working Style Graphs in ANSI Format

Working style graphs are stacked bar graphs in ANSI format. They are the graphs you would generate regularly to evaluate system performance. Working graphs are distinguished from presentation graphs, which are generated for management reports and presentations. Working graphs can be printed and displayed using conventional printers and terminals. VAX SPM produces two types of working graphs: Total and Range, specified using the TOTAL and RANGE keywords with the /STYLE qualifier.

The Total working style graph produces a stacked bar graph showing the total of one metric over time.

The Range type of working graph is usually used with the REPORT=HISTORY/CLUSTER command for cluster analysis. It graphs the minimum for a node, the maximum for a node, and the cluster average.

14.7.2.2 Specifying Presentation Style Graphs

Presentation graphs are graphs of performance data suitable for inclusion in management reports and slide presentations. These graphs can be produced in either sixel or ReGIS format.

Generate presentation quality graphs in sixel format by specifying /STYLE=SIXEL, and in ReGIS format by specifying /STYLE=REGIS.

The following command produces a default version of a presentation quality graph of CPU utilization in ReGIS format:

```
$ PERFORMANCE REPORT=LOG_FILE/GRAPH/STYLE=REGIS
```

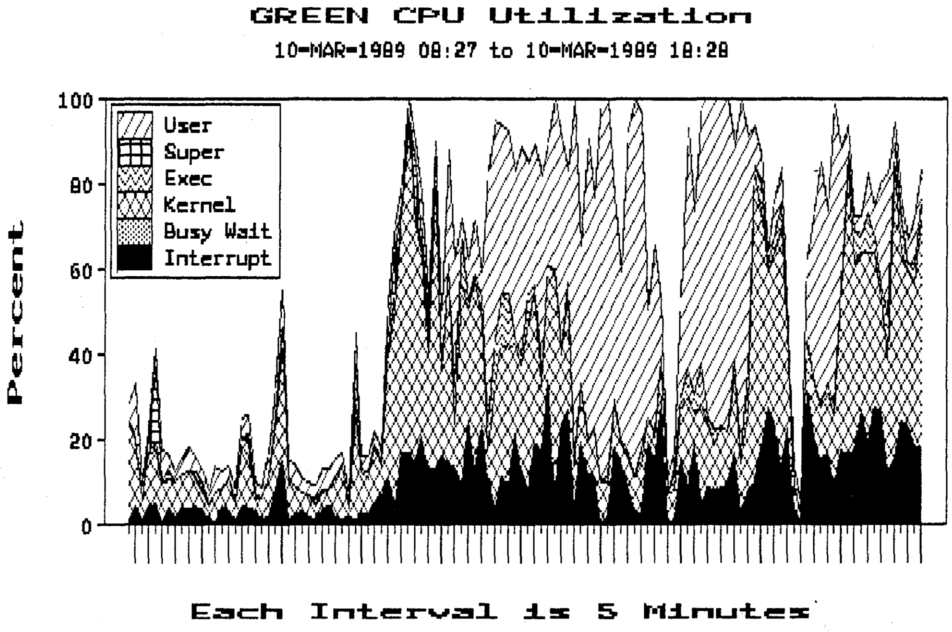
Default presentation graphs have the following characteristics:

- Graph type is a filled line graph.
- A legend is supplied, which maps colors or line types to the metrics being graphed.
- The title is supplied, which contains the following:
  - — Node or cluster name
  - — Disk or device identifier (if disk and device metrics are graphed)
  - — The graph metric name, such as CPU, Memory, or Page Faults
- The subtitle is supplied, which describes the graph reporting period.
- The horizontal label is supplied, which describes the reporting interval.

- Colors from palette number 1 are generated.
- The vertical label is supplied with the unit appropriate to the metric being graphed. For a CPU graph, the vertical label is percent (%), for a D_RATE it is rate/second, and BALANCE_SET is a number representing the number of processes.

Figure 14–5 is an example of a default presentation graph.

Figure 14–5 Example of a Default Presentation Graph



A file is created for each graph according to naming conventions described in Chapter 17. The graph generated by the above command is on node GREEN in a file named the following:

DISK$USER_A:SPM$GRAH_CPU_1.REGIS_GREEN

In addition to selecting ReGIS and sixel format, you can choose from four types of presentation quality graphs:

- Line
- Histogram
- Pie chart
- Stacked bar

The graph types can be produced using a variety of style characteristics such as color, borders, grids, and customized titles and labels. Graph types and characteristics other than the default are specified using a graph description file, which is described in Chapter 17. They can be printed and displayed using ReGIS-compatible output devices.

14.7.2.3 Requirements

To print presentation quality graphs in either ReGIS or sixel format requires ReGIS-compatible output devices such as an LN03, LCG01, LA34_VA, or LA50.

It is also possible to produce sixel files from VT240 and VT340 color terminals for printing on monochrome devices such as the LN03. To do this, set the COLOR PRINTING set-up attribute on the terminal to MONOCHROME. Failure to do this produces a solid black rectangle in place of a graph.

To display presentation quality graphs requires a ReGIS-compatible terminal such as the VT125, VT240, VT241, VT330, VT340, or PC350.

An important difference between the ReGIS and sixel graphs is the manner in which they are produced. Graphs in ReGIS format are not displayed as they are produced; therefore, they may be produced on any type of terminal. Graphs in sixel format are generated by a screen dump method in which each graph is displayed on the screen as it is produced. Therefore, sixel graphs must be generated on a ReGIS-compatible terminal such as the VT125, VT240, VT241, VT330, VT340, or PC350.

Sixel graph files can be included in VAX DOCUMENT source files.

14.7.3 Specifying Time Intervals for Plotting

By default, the time unit for plotting data is the collection interval for the log file. You can change this default using the /INTERVAL qualifier as described in Section 14.5. The time unit for plotting is a whole multiple of the log file collection interval. There are a maximum of 120 time units per graph page; if necessary, individual graphs are continued on additional pages.

14.7.4 Selective Generation of Graphs

The ability to selectively generate graphs lets you describe performance criteria to determine when graphs of disk and device statistics will be produced.

For example, if you have 50 disks on your system and you are only interested in the I/O rates for the busiest disks, you can specify that graphs be generated only for the disks that have more than 10 I/Os per second over 30% of the time.

Specifying the selective generation of graphs requires two steps:

1 Specify threshold values for graph items, devices, and disks.

2 Use the /SELECT qualifier to test the threshold values and determine whether or not to generate graphs.

Specifying the selective generation of graphs in the SPM$MANAGER.COM command procedure (Chapter 3) lets VAX SPM automatically monitor your system and generate graphs for the disks and devices on which resource utilization exceeds the thresholds you define.

14.7.4.1 Specifying Thresholds for Graphs

A threshold is a user-defined value associated with a graph. Thresholds work in conjunction with the /SELECT qualifier to determine if a graph is generated for a graph item. When the item is graphed using a threshold value, the threshold value is represented as a series of dotted lines on working style graphs.

Each graph item specified using /GRAPH can be accompanied by a threshold value. The threshold value must be a positive integer. In addition, the /DEVICE and /DISK qualifiers allow threshold values to be specified locally for each device or disk; these values override the global threshold values given with /GRAPH and its keywords.

In other words, a threshold value given with the /GRAPH qualifier can be overridden by a threshold value given for a specific device by using the /DEVICE qualifier. Likewise, a value given to a D_ keyword can be overridden by a threshold value given for a specific disk using the /DISK qualifier.

Consider how global and local threshold values are specified by the following commands:

```
$ PERFORMANCE REPORT=LOG_FILE/DEVICE=(XEA0:30,XEA1)/CLASS=DEVICE -
  /GRAPH=(DEVICE:20)/NOTABULAR SPM$COLLECT_TUNE.DAT
```

In the above command:

- /DEVICE=(XEA0:30,XEA1), specifies a threshold of 30 for device XEA0, which overrides the global threshold of 20 specified by /GRAPH=(DEVICE:20).
- Since no threshold is given for device XEA1 by /DEVICE=(XEA0:30,XEA1), the threshold of 20 specified globally by /GRAPH=(DEVICE:20) is used.
- /CLASS=DEVICE is included since device statistics are not among the default classes for reporting.
- /GRAPH with the DEVICE keyword declares that device I/O rate graphs will be generated for the devices specified by the /DEVICE qualifier.

The following command specifies threshold values for disk allocation graphs:

```
$ PERFORMANCE REPORT=LOG_FILE/GRAPH=(D_ALLOCATION:70) -
   /DISK=(DBA0:60,DBA1)/NOTABULAR LOG.DAT
```

In the above command:

- /GRAPH=(D_ALLOCATION:70) assigns a global threshold of 70 for all D_ALLOCATION items; therefore, DBA1 has a threshold value of 70.
- /DISK=(DBA0:60, DBA1) specifies that graphs will be generated for disks DBA0 and DBA1. The locally assigned threshold value of 60 for disk DBA0 overrides the globally assigned value of 70 specified by /GRAPH.

In the above commands, the qualifier /NOTABULAR excludes tabular data from these reports.

Note that in contrast to specifying disks for data collection, disk names may not contain leading underscores. If /DISK is omitted, then /DISK=ALL (for all disks in the log file) is assumed.

14.7.4.2 Using the /SELECT Qualifier

The /SELECT qualifier has two keywords, shown in Table 14–6.

Table 14–6 Keywords for the /SELECT Qualifier

Keyword	Meaning
GREATER_THAN:percent	At least "percent" of the graph columns must be greater than or equal to the threshold value specified by the /GRAPH=, /DISK, or /DEVICE qualifiers.
LESS_THAN:percent	At least "percent" of the graph columns must be less than the threshold value specified by the /GRAPH=, /DISK, or /DEVICE qualifiers.

If the keyword GREATER_THAN is specified with /SELECT, then to generate a graph, at least "percent" of the graph columns must be greater than or equal to the threshold value specified with the /GRAPH=, /DISK, or /DEVICE qualifiers. If the keyword LESS_THAN is specified with /SELECT, at least "percent" of the graph columns must be less than or equal to the threshold value specified with the /GRAPH=, /DISK, or /DEVICE qualifiers.

The GREATER_THAN and LESS_THAN keywords themselves accept a single value representing the percentage of columns graphed that must meet the threshold criteria before the graph is produced. If the GREATER_THAN keyword is specified without a value, then 30 percent is assumed. If the LESS_THAN keyword is specified without a value, then 90 percent is assumed.

If the /SELECT qualifier is present without a keyword, then GREATER_THAN:50 is assumed. If 0 is given as the percent, then only one point on the graph has to satisfy the criterion to produce the graph.

To specify that disk rate graphs be generated only for disks that have more than 10 I/Os per second over 30% of the time, you would use a command similar to the following:

```
/GRAPH=D_RATE:10/SELECT=GREATER_THAN:30
```

14.7.5 Specifying a Scaling Factor to Override Autoscaling

VAX SPM automatically establishes the scale for the vertical axis of graphs based upon the range of the units graphed. For example, if the number of page faults per seconds is in the range of 0 to 160, the increments on the vertical axis in units of 16 are labeled 16, 32, 64, ... to 160. If the number of page faults graphed is in the range of 0 to 60, the increments on the graph in units of 6 are labeled 6, 12, 24, ... to 60.

You can override the autoscaling mechanism by specifying the number you want at the top of the vertical axis as a second value to the /GRAPH, /DISK, or /DEVICE qualifier. If you specify a maximum scale value that is smaller than the actual value of the display-variable, the graph displays a carat (^) as the top line of the bar to indicate that the actual value is greater than the graph allows.

The following example specifies a threshold of 100 and a maximum scale value of 150:

```
/GRAPH=PAGE:100:150
```

If you wish to specify a maximum scale value of 50 and no threshold value, specify a threshold of 0 as in the following example:

```
/GRAPH=CPU:0:50
```

The scaling on stacked bar graphs is predetermined and therefore is not affected by scaling values specified for the /DISK, /DEVICE, and /GRAPH qualifiers.

14.8 Dumping Data Used to Generate Reports and Graphs

The contents of the data records used to generate the tabular and graph pages can be dumped in ASCII or binary format using the /DUMP qualifier. The dump file is an RMS sequential file of variable length records. The records themselves contain "computed" data, and are grouped per interval over the reporting period requested in the report or graph.

By default, the dump file is in binary (NOASCII) format and has the name SPMDUMP.DAT. Using the FILE and TYPE keywords, you can specify the dump file name and change the format to ASCII.

The /DUMP qualifier is provided for those users who want to write their own programs for reporting on performance data, or who wish to use a graphics package such as DATATRIEVE or DECgraph to format data and produce reports. DATATRIEVE record definition files are provided in the example directory and also in Appendix A. SPMASCII.DTR provides a definition for an ASCII formatted file, and SPMBINARY.DTR provides a definition for a binary (NOASCII) formatted file.

DATATRIEVE definitions may be found on-line in the VAX SPM example in SPM$EXAMPLES:.

Use the following command to dump the log file SPM$COLLECT_TUNE.DAT in binary format to the file LOG1.DMP:

```
$ PERFORMANCE REPORT=LOG_FILE -
  /DUMP=(FILE=LOG1.DMP,TYPE=NOASCII) SPM$COLLECT_TUNE.DAT
```

15 Creating and Maintaining History Files

This chapter describes the following:

- Invoking the ARCHIVE command
- Creating a history file
- Specifying a log of history file transactions
- Listing information in a history file
- Adding, replacing, and deleting records in a history file
- Merging history files

An overview of the history file data collection, archiving, and reporting procedure is shown in Figure 15–1. This chapter describes the highlighted portion of the diagram. Chapter 21 provides information relating to cluster-wide history files, which supplements the information in this chapter.

15.1 Invoking the ARCHIVE Command

To generate a history file using one or more log files created with the PERFORMANCE COLLECT=CAPACITY or PERFORMANCE COLLECT=TUNE commands, give a command of the following type:

```
PERFORMANCE ARCHIVE=HISTORY/qualifiers... hist-file [log-file,history-file...]
```

This command accepts a required parameter, the name of the history file, optional parameters in the form of one or more log or history file names, and qualifiers shown in Table 15–1. No special privileges are needed to use this command.

Figure 15–1 Collecting/Reporting Historical Data

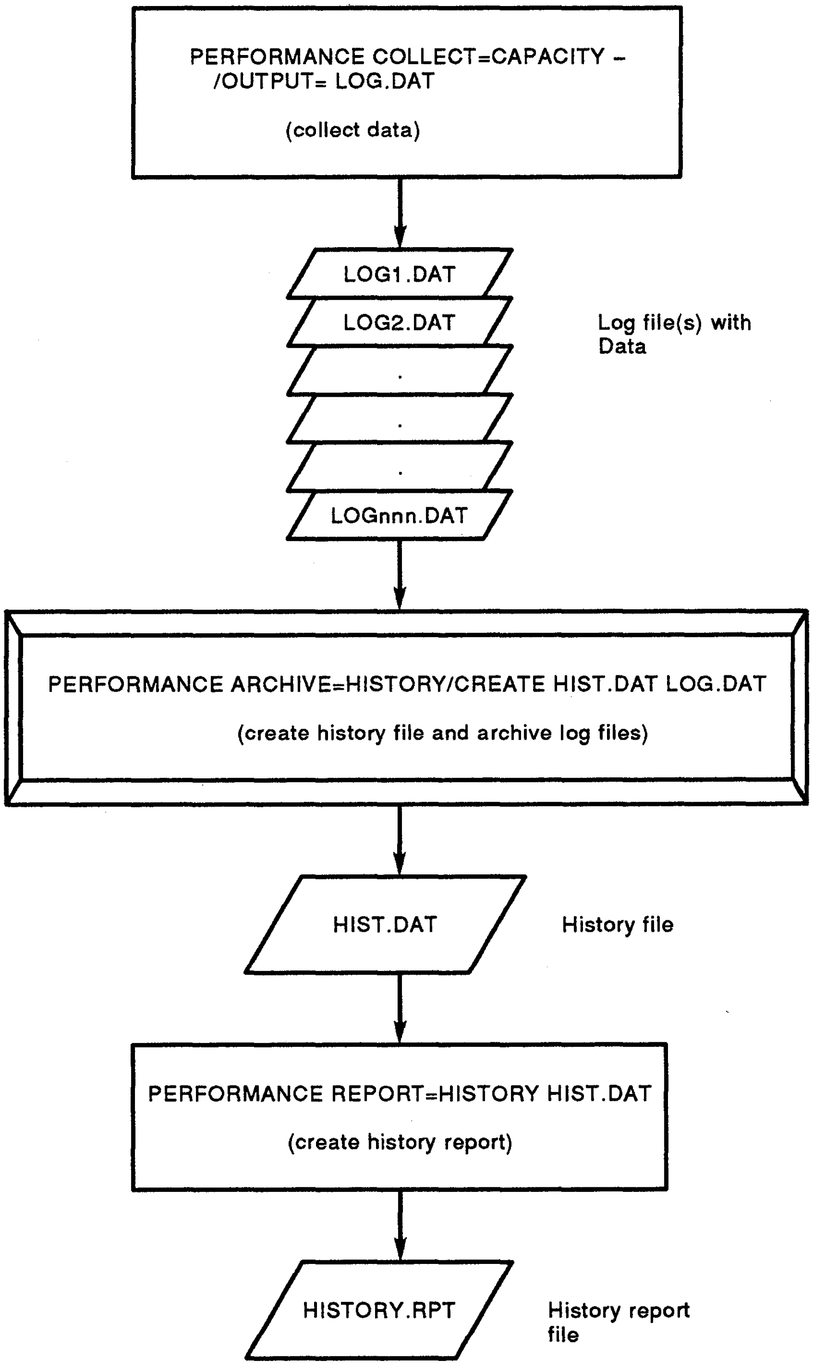

PERFORMANCE COLLECT=CAPACITY –
/OUTPUT= LOG.DAT
(collect data)
LOG1.DAT
LOG2.DAT
.
.
.
LOGnnn.DAT
Log file(s) with Data
PERFORMANCE ARCHIVE=HISTORY/CREATE HIST.DAT LOG.DAT
(create history file and archive log files)
HIST.DAT
History file
PERFORMANCE REPORT=HISTORY HIST.DAT
(create history report)
HISTORY.RPT
History report file

Table 15–1 PERFORMANCE ARCHIVE=HISTORY Command Qualifiers

Command Qualifier	Description
/[NO]ADD	Adds records to a history file. The default is ADD.
/[NO]ALLOCATION_ CLASS=(class:name, . . .)	Specifies the allocation class value to be associated with a sytem node name. The default is NOALLOCATION_CLASS.
/BEFORE[=time]	Specifies records for archiving. The default is BEFORE = end of file.
/CLASS=(item[, . . .)]	Specifies performance metrics to be archived. The default is DEFAULT_ STATISTICS.
/[NO]CREATE=([EXTEND_SIZE:blks, INITIAL_SIZE:blks])	Creates a history file of 3000 blocks with a default extent block size of 3000. The default is NOCREATE.
/[NO]DELETE	Specifies records in the history file for deletion. The default is NODELETE.
/DEVICE=([device, . . .])	Specifies devices for which data is archived. The default (DEVICE) archives data for all devices present in the log file.
/DISK=([disk, . . .])	Specifies disks for which data is archived. The default (DISK) archives data for all disks present in the log file.
/INTERVAL=interval	Specifies the archive interval of the history file. The default interval is 15 minutes.
/[NO]LIST[=type]	Specifies a listing of information for records in a history file. The default is NOLIST.
/[NO]LOG	Specifies that the date and time of each record being added, replaced, or deleted be displayed. The default is NOLOG.
/NODE_NAME=name	Specifies a node from a history file, the records of which are selected for /ADD and /REPLACE operations. With a log file, specifies the node name used during add or replace operations for supplying a missing node name or for overriding an existing node name. The default is the node name in the log file.
/[NO]REPLACE	Specifies that data records in the log file(s) whose records match those of existing records in the history file, replace the current history file records. The default is NOREPLACE.

Table 15–1 (Cont.) PERFORMANCE ARCHIVE=HISTORY Command Qualifiers

Command Qualifier	Description
/SINCE[=time]	Specifies records for archiving. The default is SINCE=beginning of file.
/[NO]SYSTEM_ID=([LOW=value, HIGH=value])	Specifies values to be used during add or replace operations to supply missing system ID information, or to select records for /LIST and /DELETE operations. The default is NOSYSTEM_ID.

15.2 Creating a History File

The ARCHIVE command provides the following options for creating a history file:

- Creating an empty history file
- Creating a history file containing log file data
- Specifying the archive interval for the history file
- Defining the initial size of the history file and the default size of its extension blocks
- Selecting records to archive based on time
- Selecting classes of data to archive
- Selecting disks and devices for archiving

Each of these options is discussed in the following sections.

15.2.1 Creating an Empty History File

Use the /CREATE qualifier of the PERFORMANCE ARCHIVE=HISTORY command to create an empty history file. In the command below, HISTORY.DAT, an empty history file of default size (3000 blocks) and default archive interval (15 minutes) is created. Data from log files can be added to the history file at a later time.

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE HISTORY.DAT
```

15.2.2 Creating a History File Containing Log File Data

Use the /CREATE qualifier with the PERFORMANCE ARCHIVE=HISTORY and specify log file names to create a history file containing log file data.

In the following command, the history file HISTORY.DAT is created with the default size (3000 blocks) and the default archive interval (15 minutes). At the same time, the default statistics (CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE) in the log files SPM$COLLECT_CAPACITY1.DAT and SPM$COLLECT_CAPACITY2.DAT are added to the history file.

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE HISTORY.DAT-
  SPM$COLLECT_CAPACITY1.DAT,SPM$COLLECT_CAPACITY2.DAT
```

Log file may be specified using the wildcard character, asterisk (*).

In the following example, the asterisk (*) is used to specify all log files having names beginning with LOG and with file types .DAT. Use of wild cards is particularly convenient when adding many versions of a log file to the history file. Note that LOG*.DAT would add LOG01.DAT, LOG02.DAT, and LOG03.DAT in sequence, but that LOG1.DAT, LOG2.DAT, and LOG10.DAT would not be added in sequence.

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE HIST03.DAT LOG*.DAT
```

15.2.3 Specifying the Archive Interval

Log files for archiving can have collection intervals in the range 1 to 3600 seconds. A history file has a much smaller number of allowed intervals for recording data. Using the /INTERVAL qualifier when creating a history file, you can choose one of the intervals described in Table 15–2 for the archive interval:

Table 15–2 History File Archive Intervals

5 minutes	1 hour (60 minutes)	8 hours (480 minutes)
15 minutes	2 hours (120 minutes)	12 hours (720 minutes)
30 minutes	4 hours (240 minutes)	24 hours (1440 minutes)

The time interval must be specified in minutes. For example, /INTERVAL=720 would give an archive interval of 12 hours.

Care must be taken when selecting the archive interval because it may be specified only once for the history file. Usually, the use of the history file determines the size of the archive interval. For example, if the log file is used for historical reporting purposes, an interval of 15, 30, or 60 minutes is adequate. A history file used for performance tuning, however, requires shorter intervals.

A long interval provides for small history files at the cost of losing information about fluctuations on a fine time scale. Likewise, a short interval provides more information about fluctutations on a fine time scale at the cost of larger history files, which require more disk space. The default archive interval is 15 minutes.

The command below creates a history file with an archive interval of 30 minutes. All data archived to this file will be stored in units of 30 minutes, regardless of the collection intervals of the log files.

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE/INTERVAL=30 HISTORY.DAT
```

Records in a history file are stored according to time units defined by the archive interval. These time units are called time buckets, or buckets. When the log file and history file intervals are different, data is averaged or prorated as it is archived into the new time bucket. History file time buckets always begin and end at definite times, including on-the-hour times. For example, if the history file interval is 15 minutes, and the log file records begin at 08:34:20, then the first history file bucket spans the time 08:30 to 08:45, the second bucket spans the time 08:45 to 09:00, and so on.

When adding data to a history file, for example when creating a history file, the creation of time buckets can be observed by specifying the /LOG qualifier as described in Section 15.3.

15.2.4 Defining the Initial Size and Extension Block Size

The /CREATE qualifier is used to create a history file. It has two keywords: EXTEND_SIZE and INITIAL_SIZE. The EXTEND_SIZE keyword specifies the default number of blocks to extend the file. The INITIAL_SIZE keyword specifies the initial size of the file in blocks.

If you are planning to create and maintain a large history file, you should specify large values for these keywords to incur less disk fragmentation as log files are added. Likewise, if you are creating a small history file, you would specify smaller values for these keywords.

The size attained by a history file depends on the following factors:

- The type of statistics collected
- The archive interval of the history file

Larger history files result from a greater number of statistics collected and a smaller archive interval. A method for estimating the size of a history file is given in Appendix F of the *VAX SPM Reference Manual*.

If you do not specify values for these keywords, both default to a value of 3000 blocks.

The command below creates a history file whose initial size is 2000 blocks. When it is necessary to extend the file, it will be extended by 2000 blocks.

```
$ PERFORMANCE ARCHIVE=HISTORY-
  /CREATE=(INITIAL_SIZE:2000,EXTEND_SIZE:2000) HISTORY.DAT
```

15.2.5 Selecting Records to Archive Based on Time

The /BEFORE and /SINCE qualifiers can be used to select, on the basis of time, which records are selected for archive operations. Both qualifiers accept absolute and delta times (see the *VMS DCL Concepts Manual* for valid time formats). If these qualifiers are omitted, all records in a log file are selected for archive operations.

The command below creates the history file HISTORY.DAT. Records are added from log file SPM$COLLECT_CAPACITY.DAT whose time stamps are before April 5, 1988 at 3 p.m.

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE-
  /BEFORE="5-APR-1988:15:00:00" HISTORY.DAT SPM$COLLECT_CAPACITY.DAT
```

The command below creates the history file HISTORY.DAT. Records are added from log file SPM$COLLECT_CAPACITY.DAT whose time stamps are later than April 6, 1984 at 9 a.m.

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE-
  /SINCE="6-APR-1984:09:00:00" HISTORY.DAT SPM$COLLECT_CAPACITY.DAT
```

These qualifiers can be used in a global fashion and also with each individual log file name, overriding the global specification.

Consider the uses of the /BEFORE and /SINCE qualifiers in this command:

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE HISTORY.DAT-
  /BEFORE=30-AUG-1988:18:00/SINCE=25-AUG-1988:8:00-
  SPM$COLLECT_CAPACITY1.DAT/BEFORE=29-AUG-1988:17:00/SINCE=29-AUG-1988:8:00-
  SPM$COLLECT_CAPACITY2.DAT
```

In the above command:

- /BEFORE=30-AUG-1988:18:00 and /SINCE=25-AUG-1988:8:00 are used globally, and determine the archive period for the log file SPM$COLLECT_CAPACITY2.DAT.
- /BEFORE=29-AUG-1988:17:00/SINCE=29-AUG-1988:8:00 are used locally, and override the global times for the log file SPM$COLLECT_CAPACITY1.DAT.

Use the /SINCE and /BEFORE qualifiers with the /ADD, /REPLACE, and /DELETE qualifiers as described in Section 15.5 to specify according to time the records to add, replace, and delete. Use the /BEFORE and /SINCE qualifiers with the /CREATE qualifier as described in Section 15.2.2 to specify according to time the records to add when creating a history file.

15.2.6 Selecting Classes of Data to Archive

Use the /CLASS qualifier to select or deselect data for archiving. All default and optional data (if present in the log file), except configuration data and process metrics, can be archived. If /CLASS is omitted from the command line, only default statistics (CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE) are archived. The items that can be archived and the keywords for specifying them are shown in Table 15–3.

Table 15–3 Keywords for the /CLASS Qualifier

Keyword	Meaning
ALL	All optional statistics
CPU	CPU statistics
DEFAULT_STATISTICS	CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_ CACHE statistics
DEVICE	Device rate/second statistics for devices specified by /DEVICE
DISK	Disk statistics for disks and servers specified by /DISK
FILE_PRIMITIVES	File system primitive counts
IO	I/O statistics
LOCK	Lock statistics
MEMORY	Memory statistics
PAGE_FAULT	Page fault statistics
SCHEDULER	Scheduler statistics
SYSTEM_ COMMUNICATION	System communication services data
XQP_CACHE	XQP caching statistics

All the above keywords are available in negated form. For example, /CLASS =(ALL, NOLOCK) will archive all statistics, including default and optional classes, except lock statistics. Note that the DEFAULT_STATISTICS class of statistics is always reported unless specifically negated. If /CLASS is omitted from the command line, then /CLASS=DEFAULT_STATISTICS is assumed. Archiving of default statistics can be disabled wholly or partially by negating and/or including specific keywords. For example, to archive only disk and CPU data, /CLASS=(NODEFAULT_STATISTICS, DISK, CPU) can be specified.

15.2.7 Selecting Disks and Devices for Archiving

Disk statistics for all disk class devices are archived by default. Data for servers for specified disks is also archived automatically. To turn off archiving of disk statistics, use the NODISK keyword with the /CLASS qualifier. To archive statistics for specified disks only, use the /DISK=(disk, . . .) qualifier.

Device statistics are archived by including the DEVICE keyword with the /CLASS qualifier. To archive specific devices rather than all devices, individual devices can be specified using the /DEVICE=(device, . . .) qualifier.

The /DISK and /DEVICE qualifiers take arguments in the form of physical device names, logical names, or truncated device names. When a truncated disk or device name is supplied, one of three actions is taken:

- If the disk or device name is truncated (for example, DB or XE), then archiving takes place for all disks or devices beginning with what was entered (in this example, DB or XE).

- If the controller designation is omitted (for example, DB2 or XE1), then archiving takes place for all disks or devices on all controllers with the specified unit number (for example, DBA2 and DBB2).
- If the unit number is omitted (for example, DBA or XEA), then archiving takes place for all disks or devices on the specified controller (for example, DBA1 and DBA2).

Disk and device names, as they appear in the log file, may be renamed during the archiving process by specifying a node name and/or allocation class using the /NODE_NAME (Section 21.2.2) and /ALLOCATION_CLASS (Section 21.2.3) qualifiers. These qualifiers cause identifying prefixes in the disk or device names to be changed.

15.3 Specifying a Log of History File Transactions

Use the /LOG qualifier with the PERFORMANCE ARCHIVE=HISTORY command to specify a log of history file transactions. The date and time of each record that is replaced, deleted, or added (by /CREATE or /ADD) is displayed.

The command below creates a new history file, HISTORY.DAT, and adds default statistics from the log file SPM$COLLECT_TUNE.DAT to it. Use of the /LOG qualifier causes the history file buckets to be displayed as they are added to. Since /INTERVAL=30 is specified, the history file archive interval is 30 minutes. Also displayed are the SCS node name and system Id for the data in each log file archived, and a message that the interval chosen is greater than the Daily reporting interval (15 minutes).

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE/LOG/INTERVAL=30 -
  HISTORY.DAT SPM$COLLECT_CAPACITY.DAT.1
%SPM-I-INTGREATER, Interval specified is greater than Daily
reporting interval
Processing file MAROON$DUA2:[SPMTEST]SPM$COLLECT_TUNE.DAT.1
Adding data for node MAROON system id hi: 3, system id lo: 4242.
Added data From :  1-APR-1988 00:00:00.00  To :  1-APR-1988 00:30:00.00
Added data From :  1-APR-1988 00:30:00.00  To :  1-APR-1988 01:00:00.00
Added data From :  1-APR-1988 01:00:00.00  To :  1-APR-1988 01:30:00.00
Added data From :  1-APR-1988 01:30:00.00  To :  1-APR-1988 02:00:00.00
```

15.4 Listing Information in a History File

Use the /LIST qualifier to list the information in a history file. The /LIST qualifier has two keywords: BRIEF and FULL. With the BRIEF keyword, the listing includes the history file archive interval, the node name and SCS system Id, and the begin/end times of continuous data in the history file. With the FULL keyword, the listing includes additional information about the statistics archived, and for a single node listing, the names of servers, disks, and devices.

In this example, the FULL keyword with the /LIST qualifier displays the history time interval; system name and Id; periods for which continuous data was found; statistics archived; and servers, disks, and devices in history file HISTORY.DAT.

```
$ PERFORMANCE ARCHIVE=HISTORY/LIST=FULL HISTORY.DAT
Data was found in history file.
History file interval is 30 minutes.
Node(s):             System id hi:  System id lo:
   MAROON                  0           7777
From :  1-APR-1988 00:00:00.00  To :  2-APR-1988 00:00:00.00
Nodes:                Archived data:
   MAROON            mem,cpu,page,io,xqp,dsk
Servers(s):           Disk(s):                Device(s):
   MBA0                  $7$DBA1
   MATENTA$              $7$DBA2
                         $7$DBA3
                         $7$DRA4
                         MAROON$DUA0
                         MAROON$DUA1
                         MAROON$DUA2
```

15.5 Adding, Replacing, and Deleting Records in a History File

Once a history file has been created as described in Section 15.2, data from log files can be added to it at any time using the PERFORMANCE ARCHIVE=HISTORY command with the /ADD qualifier. Data can also be replaced in and deleted from the history file using the /REPLACE and /DELETE qualifiers.

The /BEFORE and /SINCE qualifiers (described in Section 15.2.5) can be used to control the time ranges for records added, replaced, and deleted. Omitting these qualifiers causes the entire log file or history file to be processed.

Use the /ADD qualifier to add log file records to an existing history file.

In the command below, default statistics for all records from all versions of the log file SPM$COLLECT_CAPACITY.DAT are added to history file HISTORY.DAT.

```
$ PERFORMANCE ARCHIVE=HISTORY/ADD HISTORY.DAT SPM$COLLECT_CAPACITY.DAT.*
```

In the command below, records in SPM$COLLECT_CAPACITY.DAT whose time stamps are later than April 7, 1988 at 4 p.m. are added to the history file, HISTORY.DAT.

```
$ PERFORMANCE ARCHIVE=HISTORY/ADD-
  /SINCE="7-APR-1988:16:00:00"  HISTORY.DAT SPM$COLLECT_CAPACITY.DAT
```

When using the /ADD qualifier, if data is found that would normally be assigned to an existing bucket in the history file, the data is not added because to do so would supersede the data in the existing bucket. In that case, a message such as the following is displayed:

```
Data not superseded From :  5-APR-1988 14:15:00.00  To :  5-APR-1988 14:30:
```

Use the /REPLACE qualifier to supersede data in an existing bucket in the history file. With the /REPLACE qualifier, new buckets are added (as for the /ADD qualifier) and data in existing buckets is replaced where new records have time stamps that match the time range of existing buckets. The /LOG qualifier can be used to observe the addition/replacement process and is described in Section 15.3.

In the example below, data is added and replaced in the data buckets. The use of the /LOG qualifier produces a log of each transaction, of which only a portion is shown. Note that the log files are accessed in the order they are specified: SPM$COLLECT_CAPACITY1.DAT followed by SPM$COLLECT_CAPACITY2.DAT.

```
$ PERFORMANCE ARCHIVE=HISTORY/REPLACE/LOG HISTORY.DAT -
  SPM$COLLECT_CAPACITY1.DAT, SPM$COLLECT_CAPACITY2.DAT
Processing file SPMD$:[SPMTEST]SPM$COLLECT_CAPACITY1.DAT;1
Replacing data for node MAROON system id hi: 9, system id lo: 0007.
Replaced data From :  5-APR-1988 14:15:00.00  To :  5-APR-1988 14:30:00.00
Replaced data From :  6-APR-1988 11:30:00.00  To :  6-APR-1988 11:45:00.00
```

Use the /DELETE qualifier of the PERFORMANCE ARCHIVE=HISTORY command to delete data from a history file. Use the /BEFORE and /SINCE qualifiers to specify a range of records to be deleted. If these qualifiers are not used, all data in the history file is deleted.

```
$ PERFORMANCE ARCHIVE=HISTORY/DELETE-
  /BEFORE="9-APR-1988:00:00:00"-
  /SINCE="8-APR-1988:00:00:00" HISTORY.DAT
```

In this example, all data collected on April 8, 1988 is deleted from the history file, HISTORY.DAT.

15.6 Merging History Files

You can use the ARCHIVE command to merge history file data. This capability is useful for reporting on data from a number of history files.

Once a history file has been created as described in Section 15.2, data from other history files can be merged with it using all the ARCHIVE command qualifiers except /NODE_NAME. The /NODE_NAME qualifier works differently when merging history files than it does when merging log files.

When merging data from history files, data from all nodes contained in the history file is merged. If you wish to select records from only one node, specify that node using the /NODE_NAME qualifier.

In the following command, data from the history file HISTORY1.DAT is added to records in HISTORY.DAT:

```
$ PERFORMANCE ARCHIVE=HISTORY/ADD HISTORY.DAT HISTORY1.DAT
```

16 Reporting on History File Data

This chapter explains how to generate tabular reports and graphs from the data in a history file.

It describes the following:

- Invoking the history reporting command
- Specifying the reporting and graphing period
- Specifying prime and nonprime hours, days, and dates on which to base reports and graphs
- Specifying disks and devices for reports and graphs
- Generating daily, weekly, and monthly reports and graphs
- Defining reporting units for reports and graphs
- Specifying averaged data for reports and graphs
- Using additional reporting features
- Using additional graphing features
- Dumping the data used to generate reports and graphs
- Reporting cluster information

16.1 Invoking the History File Reporting Command

To generate reports from a history file, give a command of the following type:

```
PERFORMANCE REPORT=HISTORY/qualifiers... history-file-spec
```

This command accepts one parameter (the name of the history file) and a number of qualifiers. No special privileges are needed to use this command; however, the PGFLQUOTA quota must be greater than 8000.

Table 16–1 lists reporting qualifiers for the REPORT=HISTORY command.

Table 16–1 VAX SPM REPORT=HISTORY Command Qualifiers

Command Qualifier	Description
/AVERAGE[=type]	Used with the /DAILY, /WEEKLY, /MONTHLY, or /GENERAL qualifiers to average data over a single time period. The default is /NOAVERAGE.

Table 16–1 (Cont.) VAX SPM REPORT=HISTORY Command Qualifiers

Command Qualifier	Description
/BEFORE	Specifies a range of dates for records selected for reporting. The default is /BEFORE=end of file.
/CLASS=(item[,...])	Specifies the optional classes of statistics to be included in the tabular report or dump file. The default is DEFAULT_STATISTICS.
/[NO]CLUSTER=([NAME=cluster-name])	Specifies that a cluster analysis is to be performed. The default is /NOCLUSTER.
/[NO]DAILY[=time]	Produces a report for each day within the reporting period. The default is /DAILY=00:00:00.00.
/DEVICE=([DEVICE[:thresh:scale],...])	Specifies devices for reporting. The default is /DEVICE.
/DISK=([DISK[:thresh:scale],...])	Specifies disks for reporting. The default is /DISK.
/[NO]DUMP=([FILE=name,] [TYPE=[NO]ASCII,] [FORM=format-type])	Dumps report information into a user-specified file. The default is /NODUMP.
/[NO]GENERAL=([TITLE="title", INTERVAL=number REPORT_ INTERVAL=minutes])	Produces a report for each reporting unit defined in the file pointed to by the logical SPM$DATES. The default is /NOGENERAL.
/[NO]GRAPH=([graph-item [:thresh:scale], . . .])	Specifies the generation of graphs and selects the items to be graphed. The default is /GRAPH=SUMMARY.
/[NO]HOLIDAYS=(date[, . . .])	Specifies user-defined dates as nonprime time. The default is /NOHOLIDAYS.
/[NO]LEGEND	Generates a report page showing intervals over which data was computed for the graph or report. The default is /NOLEGEND
/[NO]MONTHLY	Produces a report for each month within the reporting period. The default is /NOMONTHLY.
/NODE=([node, . . .])	Specifies one or more VAXcluster system nodes for reporting. The default is /NODE.
/OUTPUT[=file-spec]	Specifies a report file name. The default is /OUTPUT=HISTORY.RPT. In the case of presentation quality graphs, a unique descriptive file name is generated for each graph.
/P_DAYS=([[NO]day, . . .])	Designates days of the week as prime and nonprime. The default is /P_ DAYS=(MONDAY,TUESDAY, WEDNESDAY,THURSDAY,FRIDAY, NOSATURDAY,NOSUNDAY)
/P_HOURS=([range])	Designates hours of the day as prime time. The default specifies prime hours from 8:00 a.m. to 5:00 p.m.

Table 16–1 (Cont.) VAX SPM REPORT=HISTORY Command Qualifiers

Command Qualifier	Description
/[NO]SELECT[=type:percent]	Selectively inhibits graph generation based upon user-specified threshold values. The default is /NOSELECT.
/[NO]SHIFT[=shift-type]	Specifies data for reporting based upon prime or nonprime hours. The default is /NOSHIFT.
/SINCE[=time]	Specifies a range of dates for records selected for reporting. The default is /SINCE=beginning of file.
/STYLE=type	Specifies working or presentation graphs. The default is /STYLE=TOTAL, which is a working graph in ANSI format.
/[NO]TABULAR[=(report-section[, . . .])]	Specifies the generation of a tabular report and optionally names sections to be included in the report. The default is /TABULAR=FINAL.
/[NO]WEEKLY[=([DAY=][,TIME=])]	Produces a report for each week within the reporting period. The default is /NOWEEKLY.

16.2 Specifying the Reporting and Graphing Period

Instead of reporting on data for the entire period of time covered by the history file, reports and graphs can be generated for time periods that you select. Use the /BEFORE and /SINCE qualifiers to select a period within a history file for reporting or graphing; both qualifiers accept an argument that is an absolute time, a delta time, or a combination of these two time formats.

The /SINCE qualifier specifies the date/time of the first record in the history file to be processed; if omitted, processing starts with the first record in the file. The /BEFORE qualifier specifies the date/time of the last record in the history file to be processed; if omitted, processing ends with the last record in the file. Omitting both qualifiers causes the entire history file to be processed. Since a history file can span many months, selective reporting is the usual way to report history file data.

In the following example, the reporting period begins at 8:30 a.m. on November 1, 1988 and ends at 5:30 p.m. that day:

```
$PERFORMANCE REPORT=HISTORY/SINCE="01-NOV-1988:8:30" -
/"BEFORE=01-NOV-1988:17:30" HISTORY.DAT
```

Times specified with /SINCE and /BEFORE do not have to fall within the range of dates for records in the history file; however, if no records fall within the specified range, the following message is output and the report contains no data:

```
%SPM-F-NODATAFND, No Data Found for Analysis in file
```

16.3 Specifying Prime and Nonprime Time for Reports and Graphs

The concept of prime time and nonprime time may be used to distinguish periods of peak usage of system resources from periods of lesser usage. For working graphs, prime time is indicated by the small letter "p" under the appropriate interval on the time axis. For tabular statistics, prime and nonprime time data can be generated as separate statistics pages.

The REPORT=HISTORY command provides four ways to specify prime and nonprime time for reporting as shown in Table 16–2.

Table 16–2 Qualifiers for Specifying Prime and Nonprime Time

Qualifier	Meaning
/[NO]SHIFT/=shift type	Specifies prime and/or nonprime hours for reporting or graphing
/P_HOURS	Defines hours during a day as prime hours
/P_DAYS	Defines days of the week as prime days
/HOLIDAYS	Specifies dates as nonprime days

The sections below describe each qualifier in Table 16–2.

16.3.1 Specifying Prime and Nonprime Hours for Reports and Graphs

Use the PRIME and NONPRIME keywords with the /SHIFT qualifier to select time intervals for reporting based on the concept of prime time.

Using /SHIFT=PRIME generates tabular report statistics pages and graphs that summarize data for prime hours only. Similarly, /SHIFT=NONPRIME summarizes data for nonprime hours. Since PRIME and NONPRIME cannot be specified at the same time, there are three possibilities: PRIME time reports, NONPRIME time reports, and reports in which prime and nonprime time is combined (/NOSHIFT).

As Figure 16–1 shows, each 1-hour interval during the day is designated by an integer value in the range 0 through 23. For example, the period 8 a.m. to 9 a.m. is designated by the integer 8; the period 8 a.m. to 12 p.m. is designated by the hours 8–11. By default, prime times are the hours 8–16, or 8 a.m. to 5 p.m. Times outside these times fall into the nonprime time category.

Figure 16–1 Integer Values of Hours in the Day

```
     0     1     2        7     8     9     10    11    12    13  23    24

TIME |-----|-----|----|...|-----|-----|-----|-----|-----|-----|...|-----|

HOUR    0     1    2         7     8     9     10    11    12        23
```

The definition of prime time can be changed using the /P_HOURS qualifier; this qualifier accepts integer values or ranges that override the default definition of prime time.

The /P_HOURS qualifier has the following format:

```
/P_HOURS=([range...])
```

The /P_HOURS qualifier specifies one or more ranges of hours to be prime time. A range of hours is given as two integer values separated by a hyphen (-). Multiple hours and hour ranges are separated by commas.

The following example specifies prime hours of 0800 to 1100 (8 a.m. to 11 a.m.) and 1300 to 1800 (1 p.m. to 6 p.m.).

```
/P_HOURS=(8-10,13-17)
```

If the /P_HOURS qualifier is omitted from the command line, the default of /P_HOURS=(8–16) (8 a.m. to 5 p.m. as prime time) is assumed.

16.3.2 Specifying Prime and Nonprime Days for Reports and Graphs

Use the /P_DAYS qualifier to designate days of the week as prime and nonprime days.

The /P_DAYS qualifier has the following format:

```
/P_DAYS=([NO]day[,...])
```

It defines days of the week as having prime hours, or as nonprime time. By default, MONDAY through FRIDAY have prime hours, while SATURDAY and SUNDAY are nonprime time. Days can be MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY.

In the example below, Monday is nonprime time, whereas Saturday can have prime hours. Other days of the week take their default values, so TUESDAY through SATURDAY can have prime hours, while SUNDAY and MONDAY are nonprime time.

```
/P_DAYS =(NOMONDAY,SATURDAY)
```

In addition to specifying days of the week as nonprime, you can also specify dates as nonprime. There are two ways of specifying nonprime dates: by supplying dates to the /HOLIDAYS qualifier, and by supplying dates in a file pointed to by the logical SPM$HOLIDAYS.

16.3.2.1 Specifying Nonprime Dates Using /HOLIDAYS

Use the /HOLIDAYS qualifier when you have only a few dates that you want to designate as nonprime. The /HOLIDAYS qualifier has the following format:

```
/HOLIDAYS=(date[,...])
```

With the /HOLIDAYS qualifier, you can specify single date or a range of dates as nonprime.

Specify single dates separated by commas using the following format:

```
dd-mmm-yyyy
```

The following example specifies single date values for the /HOLIDAYS qualifier:

```
/HOLIDAYS=(02-JAN-1989,29-MAY-1989)
```

Specify a range of dates as two dates separated by commas and enclosed by quotation marks. When specifying a range of dates, you have the option of specifying hours in the day as nonprime. Ranges of dates have the following format:

```
dd-mmm-yyyy:hh:mm:ss
```

The example below specifies the afternoon preceding Thanksgiving to the following Saturday as nonprime, as well as Christmas Day:

```
/HOLIDAYS=("22-NOV-1989:12:00,25-NOV-1988",25-DEC-1989)
```

16.3.2.2 Specifying Nonprime Dates Using a Holidays File

Specify nonprime dates using a holidays file when you want to specify a number of dates as nonprime; for example, holidays for two years. You can specify up to 60 dates in the holidays file.

A holidays file is an ASCII file pointed to by the logical SPM$HOLIDAYS. As with the /HOLIDAYS qualifier, you can specify dates, date ranges, and times within the date ranges as nonprime. Dates are specified in the same format as with the /HOLIDAYS qualifier, one date to a line. Date ranges are specified one to a line, separated by commas.

An attempt to translate the SPM$HOLIDAYS logical is made whether or not the /HOLIDAYS qualifier is present in the command line. Therefore, once the logical is defined to point to a file, dates in that file will always be treated as nonprime days for reporting and graphing.

The following is an example of holidays for 1988 and 1989 in a holidays file. Note that ranges of dates are specified for Thanksgiving for both years. These ranges specify the afternoon preceding Thanksgiving as nonprime.

```
01-JAN-1988
30-MAY-1988
04-JUL-1988
05-SEP-1988
25-NOV-1988:12:00,27-NOV-1988
25-DEC-1988
27-DEC-1988
02-JAN-1989
29-MAY-1989
03-JUL-1989
04-JUL-1989
04-SEP-1989
22-NOV-1989:12:00,24-NOV-1989
25-DEC-1989
```

16.4 Specifying Disks and Devices for Reports and Graphs

Use the /DEVICE and /DISK qualifiers to select those devices and disks that are to be included in the graph and tabular sections of the report or the dump file. Device and disk names must be physical names, but it is not necessary to include a node prefix if one is present in the history file. Physical names can be truncated to specify a group of devices or disks. When a truncated disk or device name is supplied, one of three actions is taken:

- If the disk or device name is truncated (for example, DB or XE), then reporting is done for all disks or devices beginning with what was entered (in this example, DB or XE).
- If the controller designation is omitted (for example, DB2 or XE1), then reporting is done for all disks or devices on all controllers with the specified unit number (for example, DBA2 and DBB2).
- If the unit number is omitted (for example, DBA or XEA), then reporting is done for all disks or devices on the specified controller (for example, DBA1 and DBA2).

By default, all devices and disks will be graphed if an applicable graph item is specified using /GRAPH. The keywords beginning with D_ (for example, D_UTILIZATION) are the graph items that apply to disks, and cause disk data such as storage allocation, I/O rate, response time, and utilization to be plotted. The D_ keywords can be accompanied by a threshold value as described in Section 16.9.4.1.

To specify individual devices and disks, use the /DEVICE and /DISK qualifiers in addition to /GRAPH. For example, specifying /GRAPH=DEVICE/DEVICE=TTA6 causes the device I/O rate for device TTA6 to be plotted. The DEVICE keyword can also be accompanied by a threshold value as described in Section 16.9.4.1.

Although by default all devices and disks will be graphed, Section 16.9.4 describes how to inhibit the generation of graphs for devices and disks based upon thresholds you define.

The command in the following example would generate tabular, graphic disk utilization, and rate statistics for all the DU disks (for example, $255DUA0, DUA0, and DUA1).

```
$ PERFORMANCE REPORT=HISTORY/DISK=DU/GRAPH=(D_UTILIZATION,D_RATE)/TABULAR -
  HISTORY.DAT
```

16.5 Generating Daily, Weekly, or Monthly Reports and Graphs

A reporting unit is the measure of time for which statistics are reported or graphed. You can specify reporting units of a day, a week or a month using the /DAILY, /WEEKLY, and /MONTHLY qualifiers. You can also define your own reporting units using the /GENERAL qualifier described in Section 16.6. If none of the above qualifiers is specified, the /DAILY qualifier is the default.

The default interval and reporting unit for the daily, weekly, and monthly qualifiers are given in Table 16–3.

Table 16–3 Default Interval and Reporting Unit

Qualifier	Default Reporting Interval	Default Reporting Unit
/DAILY	15 minutes	24 hours (1 day)
/WEEKLY	2 hours	7 days (1 week)
/MONTHLY	8 hours	Up to 35 days

When generating graphs or reports using the /DAILY, /WEEKLY, and /MONTHLY qualifiers, there are default times for beginning and ending a single graph or report. These default times are shown in Table 16–4.

Table 16–4 Default Beginning and Ending Times for Reporting Units

Reporting Unit Qualifier	Default Beginning and Ending Times
/DAILY	Begins at 00:00 and ends at 24:00 hours.
/WEEKLY	Begins at 00:00 on Sunday and ends 7 full 24-hour days later.
/MONTHLY	Begins at 00:00 on the first day of the month and ends at 00:00 hours on the first day of the following month. For example: `The month of September begins on 01-SEP-1988:00:00 and ends 01-OCT-1988:00:00`

The default beginning and ending times for /DAILY and /WEEKLY can be changed using keywords and values with these qualifiers. For /MONTHLY, defaults can be changed by placing beginning and ending dates in a file pointed to by logical name SPM$DATES.

In the example below, a day begins at 8 a.m. and ends 24 hours later. You can specify the beginning time as an absolute time, a delta time, or as a combination of these two types. A beginning time is adjusted back to the nearest 15-minute interval (for example, 08:09 is treated as 08:00).

```
/DAILY=08:00
```

In the example below, the week begins at 8:00 a.m. on Monday morning and ends 7 full 24-hour days later. Note that the given time is adjusted back to the nearest 2-hour interval.

```
/WEEKLY=(DAY=MONDAY,TIME=08:35)
```

If the logical name SPM$DATES is defined to point to an ASCII text file and /MONTHLY is specified, the command takes the starting and ending dates and times of a month(s) from this file. Each line in the file has this format:

```
dd-mmm-yyyy hh:mm:ss.cc,dd-mmm-yyyy hh:mm:ss.cc
```

To extend the month of JULY by two days for reporting purposes, the file pointed to by SPM$DATES may contain dates as in the following example:

```
01-JUL-1989 00:00:00.00,03-AUG-1989 00:00:00.00
```

Months of up to 35 days can be specified. In the above example, note that the beginning time is adjusted back to the nearest 8-hour interval while the ending time is adjusted forward to the nearest 8-hour interval.

System metrics in graph form are reported as individual graphs in which values are plotted against time. Unless the /GENERAL qualifier is specified, reporting units shown in Table 16–3 are the total time represented by the axis on a single graph and the default interval is the time unit represented by each column. As previously noted, the /BEFORE and /SINCE qualifiers determine the reporting period; since the report period can contain a number of report units, there can be as many graphs in a report (each on a separate output page) as are needed to span the reporting period. For example, if there are 2 years of data in your history file and you specified /MONTHLY without specifying a reporting period, each column in the graph would represent 8-hour intervals, and a graph would be generated for each month represented in the history file.

System metrics in tabular form may be presented in individual (Interval) reports or in a single (Final) report for the reporting unit and reporting period specified. If no reporting unit is specified or defined, /DAILY is the default. When /DAILY and /TABULAR=FINAL are specified, a final report is generated for each day in the reporting period. When /DAILY and /TABULAR=INTERVAL are specified, an Interval report is generated for each 15-minute interval within the reporting period. Table 16–5 shows how interval and final tabular statistics are reported for each reporting unit.

Table 16–5 Output for Interval and Final Tabular Reports for Reporting Units

Reporting Unit Qualifier	/TABULAR=INTERVAL	/TABULAR=FINAL
/DAILY	1 report page for each 15-minute interval	1 report for each 24-hour day
/WEEKLY	1 report page for each 2-hour interval	1 report for each 7-day week
/MONTHLY	1 report page for each 8-hour interval	1 report for each month

In the example below, final statistics are tabular values computed over the week beginning June 2 at 8 a.m. A single tabular page is generated. Graph pages are suppressed using the /NOGRAPH qualifier.

```
$ PERFORMANCE REPORT=HISTORY/SINCE="02-JUN-1986 08:00" -
  /BEFORE="09-JUN-1986 08:00"/WEEKLY=(DAY=MONDAY,TIME=08:00) -
  /TABULAR=(FINAL)/NOGRAPH HISTORY.DAT
```

In the following example, the INTERVAL keyword with the /TABULAR qualifier causes tabular pages to be generated for each 2-hour (/WEEKLY) interval beginning at 8 a.m. on June 2, 1986, and continuing until June 9, 1986 at 8 a.m.

```
$ PERFORMANCE REPORT=HISTORY/SINCE="02-JUN-1986 08:00" -
  /BEFORE="09-JUN-1986 08:00"/WEEKLY=(DAY=MONDAY,TIME=08:00) -
  /TABULAR=(INTERVAL)/NOGRAPH HISTORY.DAT
```

16.6 Defining Reporting Units and Intervals for Reports and Graphs

A reporting unit is a measure of time for which statistics are reported or graphed. The reporting interval is the frequency with which data is reported within each reporting unit. For reporting purposes, the value of the report interval as shown in Table 16–3 is usually contingent upon the reporting unit. The /GENERAL qualifier allows you to specify reporting units and reporting intervals other than those provided by the /DAILY, /WEEKLY, and /MONTHLY qualifiers described in Section 16.5. For example, you can specify fiscal quarters or months for which to report and graph statistics.

You can use the /GENERAL qualifier to specify the reporting units and intervals used to calculate statistics in one of two ways:

- You can define reporting units and intervals in an ASCII text file pointed to by the logical name SPM$DATES. Each date in the text file would represent a reporting interval, and the reporting period described by all the dates in the file would represent the reporting unit. In addition, a reporting interval smaller than the one represented by each date may be specified as a value to the INTERVAL keyword.
- You can specify a reporting interval as a value in minutes for the REPORT_INTERVAL keyword.

The /GENERAL qualifier allows the most flexible method of reporting on historical reporting data. Since the user has complete control over the time intervals for reporting and graphing, the /GENERAL qualifier permits true financial reporting adapted to the user's own scheme for financial analysis.

The /GENERAL qualifier has the following format:

```
/GENERAL=([TITLE="title",INTERVAL=number, -
 REPORT_INTERVAL=minutes])
```

The "title" may be an ASCII string of up to 80 characters, and is used as part of the title for all graphs and tabular reports. The INTERVAL keyword specifies a reporting interval when the logical SPM$DATES points to a dates file. The REPORT_INTERVAL keyword specifies the reporting interval when no dates file is used.

16.6.1 Specifying Reporting Units and Intervals Using a Dates File

Each line in a dates file consists of a starting and ending date and time as follows:

```
dd-mmm-yyyy hh:mm:ss.cc,dd-mmm-yyyy hh:mm:ss.cc
```

The beginning time of a specified interval is adjusted back to the nearest archive interval, while the ending time is adjusted forward to the nearest archive interval. Time intervals represented by each line in the file need not be contiguous, but must be in ascending order. Data that falls in each time interval is combined to produce a single point for plotting on a graph or for a single interval tabular report page. In other words, one line in the dates file generates one point (column) on a graph or one interval tabular report section.

The maximum number of points (columns) per graph is 105; this is also the default value. If the dates file contains more than 105 lines, more than one graph is produced [unless /AVERAGE is used (Section 16.7)]. The number of columns per graph can be controlled using the INTERVAL keyword with the /GENERAL qualifier.

The INTERVAL keyword may be used in conjunction with the dates file, and controls the number of time intervals or points (columns) to be contained in a single graph. The value "number" may range from 1 to 105. For example, if INTERVAL=10, there will be only 10 points or columns on a single graph page, and the units on the time axis will be chosen so as to space these points evenly across the graph page. There will be as many graph pages as needed to graph all points in the reporting period.

Use the INTERVAL keyword to specify an interval other than the default specified by the number of dates (lines) in the SPM$DATES file. Unless you are using the /AVERAGE qualifier as described in Section 16.7, take care when selecting a value for the INTERVAL keyword. The value of INTERVAL must be equal to or a multiple of the archive interval for the history file, and it must be appropriate to the reporting unit defined in the dates file. For example, if you are defining weeks in a fiscal quarter in your dates file, and you want each column in the graph to represent a day, you would specify /INTERVAL=98. (The number of weeks in the quarter times the number of days in the week, 14 × 7). The graph would have 98 columns, each representing statistics for one day. If you did not specify a value for the INTERVAL keyword, each column in the graph would represent a week.

For tabular reports, the INTERVAL number also determines how many reporting intervals are reported in a final statistics page. If INTERVAL=15, then 15 reporting intervals (from the SPM$DATES file, or as given by the REPORT_INTERVAL keyword) will be combined to produce a value for a statistic in the final report.

To generate a graph showing process count (the number of processes on the system) for weeks in a fiscal quarter, you would perform the following steps:

1. Create a file named FISCALQ.TXT containing dates that define the beginning and ending dates of weeks in the quarter as shown in Figure 16–2.
2. Define the logical SPM$DATES to point to the file containing the dates as in the following example:

   ```
   $ ASSIGN FISCALQ.TXT SPM$DATES
   ```

3. Type the following command:

   ```
   $ PERFORMANCE REPORT=HISTORY/NOTABULAR/GRAPH=(PROCESS_COUNT) -
     /GENERAL=(TITLE="Q4 OF FY89") [SPM.HISTORY]SPM$HISTORY.DAT
   ```

   In the above command:

   The /NOTABULAR qualifier suppresses the generation of the tabular report.

   The /GRAPH qualifier specifies process count metrics to be graphed.

   The /GENERAL qualifier specifies that time intervals for computing data are taken from the file pointed to by the logical name SPM$DATES.

Figure 16–2 Example of SPM$DATES File for Fiscal Quarter

```
27-MAR-1989,02-APR-1988
03-APR-1989,09-APR-1988
10-APR-1989,16-APR-1988
17-APR-1989,23-APR-1988
24-APR-1989,30-APR-1988
01-MAY-1989,07-MAY-1989
08-MAY-1989,14-MAY-1989
15-MAY-1989,21-MAY-1989
22-MAY-1989,28-MAY-1989
29-MAY-1989,04-JUN-1989
05-JUN-1989,11-JUN-1989
12-JUN-1989,18-JUN-1989
19-JUN-1989,25-JUN-1989
26-JUN-1989,02-JUL-1989
```

Points for the number of date ranges specified in FISCALQ.TXT are plotted on a single graph page. As many as 105 points (columns) can be plotted on a single graph.

Figure 16–3 shows the graph page generated. The Legend page (not shown) reproduces (in different format) the contents of the dates file. On the graph page, numbers along the x axis correspond to the numbered intervals on the Legend page. The Notes on the graph page direct the user to the corresponding Legend or Report Interval page.

Figure 16–3 Process Count for Fiscal Quarter Using /GENERAL

```
Report Date:  6-MAR-1989 13:17                                 GREEN                         VAX SPM V3.3           Page     1

                                                   Q4 of FY89
                                              Process Count (Average)

       Graph From: 27-MAR-1989 00:00:00.00                              Graph To: 02-JUL-1989 00:00:00
       -------+------+------+------+------+------+------+------+------+------+------+------+------+-------
   60  +                                                                                                  +       Averages
       !                                                                                                  !     (% Monitored)
   54  +                                                                                                  +
       !                                                                                                  !
   48  +                                                                                                  + Network           6
       !                                                                                                  ! Batch             0
   42  +              I                                                                                   + Other            14
       !              I                                                                                   ! Interactive      10
   36  +       I      I                                                                                   +
       !       I      I      I      I                                                                     !
   30  +       I      I      I      I      I                                                              +
       !       I      I      I      I      I      I             I      I      I      I      I      I      !
   24  +       I      I      I      I      I      I      I      I      I      I      I      I      I      +
       !       O      O      I      I      I      I      O      O      O      O      O      O      O      !
   18  +       O      O      O      O      O      O      O      O      O      O      O      O      O      +
       !       O      O      O      O      O      O      O      O      O      O      O      O      O      !
   12  +       O      O      O      O      O      O      O      O      O      O      O      O      O      +
       !       N      N      O      O      O      O      O      O      O      O      O      O      O      !
    6  +       N      N      N      N      N      N      N      N      N      N      N      N      N      +
       !       N      N      N      N      N      N      N      N      N      N      N      N      N      !
       -------+------+------+------+------+------+------+------+------+------+------+------+------+-------
              p      p      p      p      p      p      p      p      p      p      p      p      p
              1      2      3      4      5      6      7      8      9     10     11     12     13

                                         (p = column includes prime time)
                       Legend                                                              Prime Hours
                       ------                                                              -----------
       "N" = Network           "B" = Batch                                           0800-1700
       "O" = Other             "I" = Interactive
       "." = Data Unavailable


                                                                                           Prime Days
                                                                                           ----------
                                                                                       MON,TUE,WED,THU,FRI

       Notes:
                The graph AVERAGE is at 30, with a MINIMUM value of 23, and a MAXIMUM value of 43.
```

16.6.2 Specifying the Reporting Interval Using REPORT_INTERVAL

Use the REPORT_INTERVAL keyword when no dates file is present, to specify a reporting interval that is shorter than the one associated with the reporting unit you are using. For example, if you are producing daily reports and you want to graph data for 5-minute intervals rather than 15, specify REPORT_INTERVAL=5. When selecting a value for the REPORT_INTERVAL keyword, be sure to select a reporting interval that is longer than or equal to the archive interval of the history file in order to avoid reporting meaningless data.

The value you specify for REPORT_INTERVAL must be equal to or a multiple of the history file archive interval. If it is not a multiple of the history file archive interval, the REPORT_INTERVAL value is always adjusted to the next higher whole multiple of the history file archive interval. If an SPM$DATES file is present, the REPORT_INTERVAL keyword value is ignored. If there is no SPM$DATES file present and the REPORT_INTERVAL keyword is not specified, the archive interval of the history file is used as the basis for reporting and graphing.

16.7 Specifying Averaged Data for Reports and Graphs

When data in a history file is graphed or reported for a day, week, or month, that data is prorated or combined to represent statistics appropriately. Sometimes, however, it is useful to average data over a day, week, or month.

Use the /AVERAGE qualifier to average graph or tabular data. Data within the time period specified by /SINCE and /BEFORE is averaged together as a single time period based on the /DAY, /WEEK, /MONTH, or /GENERAL qualifiers. The /AVERAGE qualifier has the following format:

```
/AVERAGE
```

Unless the /SHIFT qualifier is also used with the keywords PRIME or NONPRIME, all data records that fall in the time range specified by /SINCE and /BEFORE will be analyzed.

For example, suppose the /SINCE and /BEFORE qualifiers specify a 5-week period of time, and the /WEEKLY qualifier is used. The report will contain a single weekly graph, which is the average of five weekly graphs that would have been generated if /AVERAGE had been omitted. Corresponding time intervals (for example, 0800 to 1000 on Monday for each of the five weeks) are averaged together to produce a single graph point for 0800 to 1000 on Monday, and so on for the other graph points. The Legend page lists and group together all the intervals that were averaged to produce each graph point.

If /AVERAGE is used with the /GENERAL qualifier, the user can control which time periods are averaged together in the following ways:

- Using the SPM$DATES file
- Using the REPORT_INTERVAL and INTERVAL keywords

For example, if the INTERVAL keyword specifies 20 points (columns) per graph, and the SPM$DATES file contains 100 time intervals, then intervals 1, 21, 41, 61, and 81 will be averaged together to produce the first point on the graph. The second graph point will be the average of intervals 2, 22, 42, 62, and 82, and so on for the other points. A single graph containing 20 points or columns is generated.

In the example below, daily graphs (/DAILY is assumed) for May 6, 1988 through May 13, 1988 are averaged to produce a single daily summary graph and a single daily balance set graph. Both prime and nonprime time is included when averaging data to produce each graph point.

```
$ PERFORMANCE REPORT=HISTORY/SINCE="06-MAY-1988" -
  /BEFORE="13-MAY-1988"/AVERAGE -
  /GRAPH=(SUMMARY,BALANCE_SET)/NOTABULAR SPMHISTORY.DAT
```

Figure 16–4 is an example of a system summary graph showing a typical day for this period.

In the example below, data for time intervals specified in an SPM$DATES file is averaged together. The dates file contains date ranges for 28 days from June 5, 1988 to July 3, 1988 (for example, 05-JUN-1988 to 06-JUN-1988, 06-JUN-1988 to 07-JUN-1988, and so on). The value of 7 for the INTERVAL keyword with the /GENERAL qualifier specifies 7 points (columns) per graph. Therefore, intervals 1, 8, 15, . . . and so on are averaged together to generate the first graph point. This produces points for an average Sunday (point 1), an average Monday (point 2), and so on, for the 4 weeks in June, 1988.

```
$ ASSIGN DATES.DAT SPM$DATES
$ PERFORMANCE REPORT=HISTORY/AVERAGE -
  /GENERAL=(INTERVAL=7)/GRAPH=(SUMMARY,BALANCE_SET) -
  /NOTABULAR SPMHISTORY.DAT
```

Figure 16–5 is an example of a balance set count graph of a typical week between June 5 and July 3, 1988.

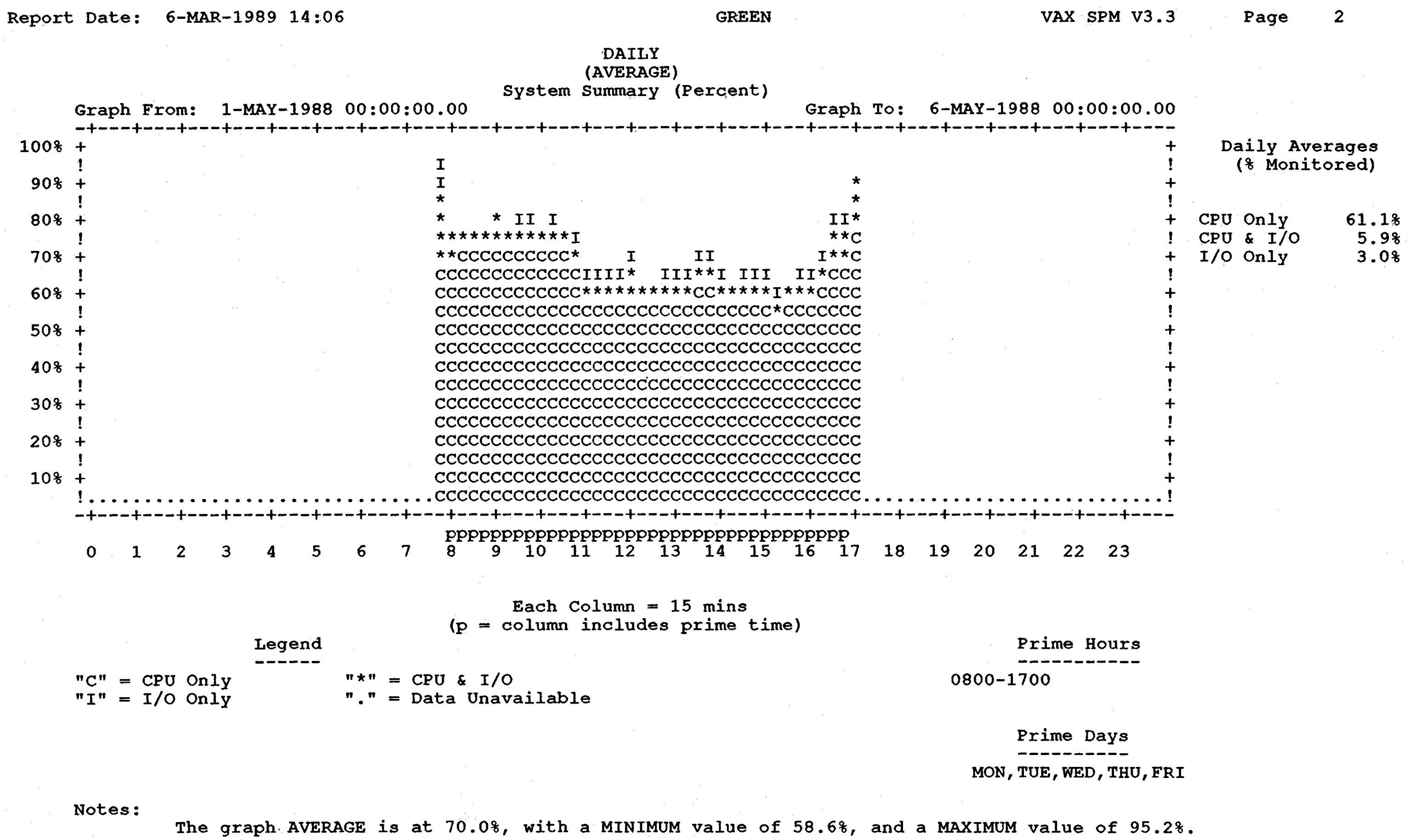
16.8 Additional Reporting Features

Use the /TABULAR qualifier with the PERFORMANCE REPORT=HISTORY command to generate tabular reports.

In addition to the reporting capabilites described in previous sections, this section describes the following:

- Reporting on various classes of performance metrics
- Producing interval reports to report performance metrics for each interval and producing final reports for entire reporting periods
- Producing interval and final reports for prime and nonprime hours and days

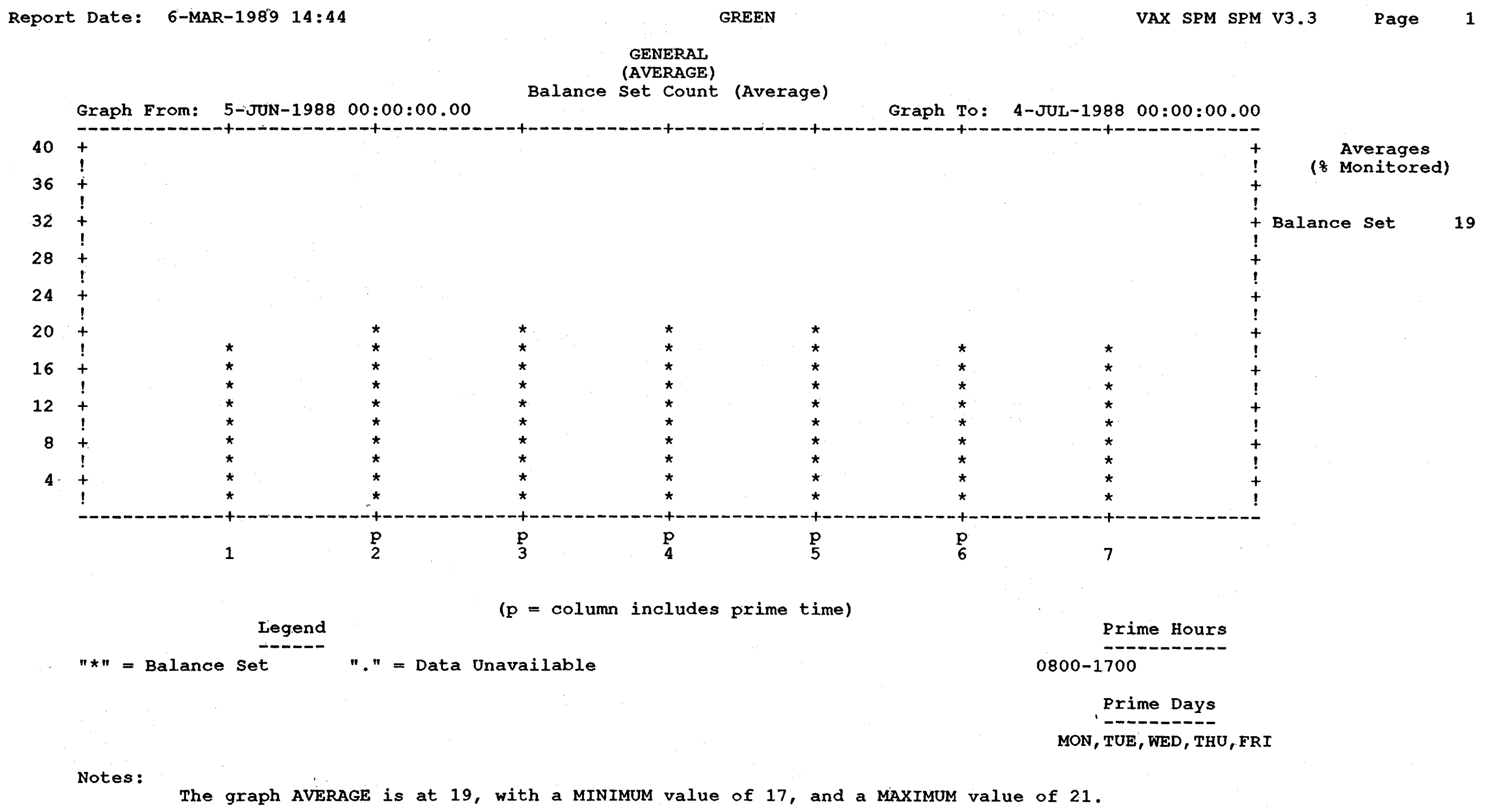
Read the *VAX SPM Reference Manual* for a full description of tabular reports and their contents.



```
Report Date:  6-MAR-1989 14:06                                 GREEN                        VAX SPM V3.3      Page    2

                                                     DAILY
                                                   (AVERAGE)
                                            System Summary (Percent)
      Graph From:  1-MAY-1988 00:00:00.00                              Graph To:  6-MAY-1988 00:00:00.00
      -+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+----
 100% +                                                                                                +    Daily Averages
      !                                I                                                               !      (% Monitored)
  90% +                                I                                    *                          +
      !                                *                                    *                          !
  80% +                                *    * II I                        II*                          +  CPU Only      61.1%
      !                                ************I                      **C                          !  CPU & I/O      5.9%
  70% +                                **CCCCCCCCCC*    I     II         I**C                          +  I/O Only       3.0%
      !                                CCCCCCCCCCCCCIIII*  III**I III  II*CCC                          !
  60% +                                CCCCCCCCCCCCC**********CC*****I***CCCC                          +
      !                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC*CCCCCCC                          !
  50% +                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                          +
      !                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                          !
  40% +                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                          +
      !                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                          !
  30% +                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                          +
      !                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                          !
  20% +                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                          +
      !                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                          !
  10% +                                CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                          +
      !................................CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC..........................!
      -+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+----
                                        PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
       0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23


                                             Each Column = 15 mins
                                        (p = column includes prime time)
                      Legend                                                            Prime Hours
                      ------                                                            -----------
      "C" = CPU Only          "*" = CPU & I/O                                       0800-1700
      "I" = I/O Only          "." = Data Unavailable

                                                                                          Prime Days
                                                                                          ----------
                                                                                      MON,TUE,WED,THU,FRI

      Notes:
               The graph AVERAGE is at 70.0%, with a MINIMUM value of 58.6%, and a MAXIMUM value of 95.2%.
```

Figure 16–4 An AVERAGE DAILY graph

Figure 16–5 An AVERAGE GENERAL graph



```
Report Date:  6-MAR-1989 14:44                                 GREEN                                VAX SPM SPM V3.3     Page    1
                                                       GENERAL
                                                      (AVERAGE)
                                              Balance Set Count (Average)
      Graph From:  5-JUN-1988 00:00:00.00                                     Graph To:  4-JUL-1988 00:00:00.00
      -------------+------------+------------+------------+------------+------------+------------+-------------
  40  +                                                                                                       +       Averages
      !                                                                                                       !    (% Monitored)
  36  +                                                                                                       +
      !                                                                                                       !
  32  +                                                                                                       + Balance Set     19
      !                                                                                                       !
  28  +                                                                                                       +
      !                                                                                                       !
  24  +                                                                                                       +
      !                                                                                                       !
  20  +                         *            *            *            *                                      +
      !            *            *            *            *            *            *            *            !
  16  +            *            *            *            *            *            *            *            +
      !            *            *            *            *            *            *            *            !
  12  +            *            *            *            *            *            *            *            +
      !            *            *            *            *            *            *            *            !
   8  +            *            *            *            *            *            *            *            +
      !            *            *            *            *            *            *            *            !
   4  +            *            *            *            *            *            *            *            +
      !            *            *            *            *            *            *            *            !
      -------------+------------+------------+------------+------------+------------+------------+-------------
                                p            p            p            p            p
                   1            2            3            4            5            6            7

                                           (p = column includes prime time)
                      Legend                                                                   Prime Hours
                      ------                                                                   -----------
      "*" = Balance Set       "." = Data Unavailable                                       0800-1700

                                                                                               Prime Days
                                                                                               ----------
                                                                                           MON,TUE,WED,THU,FRI

      Notes:
               The graph AVERAGE is at 19, with a MINIMUM value of 17, and a MAXIMUM value of 21.
```

16.8.1 Specifying the Metrics for a Tabular Report

Use the /CLASS qualifier with the /TABULAR qualifier to control the classes of statistics that appear on a tabular report. The /CLASS qualifier has the following format:

```
/CLASS=(item,[...])
```

By using the /CLASS qualifier, all default and optional data present in the history file can be reported. Keywords for the /CLASS qualifier are described in Table 16–6.

If /CLASS is omitted from the command line, only default statistics (CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE) are reported.

Table 16–6 Keywords for the /CLASS Qualifier

Keyword	Meaning
ALL	All optional statistics
CPU	CPU statistics
DEFAULT_STATISTICS	CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE statistics
DEVICE	Device rate/second statistics for devices specified by /DEVICE
DISK	Disk statistics for disks specified by /DISK
FILE_PRIMITIVES	File system primitive counts
IO	I/O statistics
LOCK	Lock statistics
MEMORY	Memory statistics
PAGE_FAULT	Page fault statistics
SCHEDULER	Scheduler statistics
SYSTEM_COMMUNICATION	System communication services data
XQP_CACHE	XQP caching statistics

All the above keywords are available in negated form. For example, /CLASS =(ALL,NOLOCK) will include all statistics, default and optional classes, except lock statistics. Note that the DEFAULT_STATISTICS class of statistics is always reported unless specifically negated. Reporting of default statistics can be disabled wholly or partially by negating and/or including specific keywords. For example, to report only disk and CPU data, /CLASS=(NODEFAULT_STATISTICS, DISK,CPU) can be specified.

In the example below, default statistics and lock statistics for the history file HISTORY.DAT are reported. Note that the default statistics CPU, DISK, IO, MEMORY, PAGE_FAULT, and XQP_CACHE are always reported unless specifically negated.

```
$ PERFORMANCE REPORT=HISTORY/CLASS=LOCK/TABULAR=FINAL -
 /NOGRAPH  HISTORY.DAT
```

In the following example, default statistics and all optional statistics (except lock statistics) for HISTORY.DAT are reported:

```
$ PERFORMANCE REPORT=HISTORY/CLASS=(ALL,NOLOCK) -
 /NOGRAPH HISTORY.DAT
```

Note that in the above examples the /NOGRAPH qualifier suppresses the generation of graphs.

16.8.2 Specifying the Type of Tabular Report

The /TABULAR qualifier controls not only the presence of tabular data in a report, but also the intervals for reporting data. With regard to time, there are two types of reporting (final statistics and interval statistics) as follows:

1 Final statistics are for the entire reporting period. Each item of tabular data appears just once in the report. Figure 16–6 shows the first page of a Final Statistics report.

2 Interval statistics are for a specific interval. Each item of tabular data appears as many times as there are intervals in the reporting period; therefore, Interval Statistics reports may produce a significant amount of output.

Keywords used with /TABULAR that are common to the REPORT=HISTORY command are shown in Table 16–7.

Table 16–7 Keywords for the /TABULAR Qualifier

Keyword	Meaning
ALL	All keyword options shown below
BYCLUSTER	A single report that summarizes data from all nodes and reports it as a single unit
BYNODE	Reports each node's contribution to the cluster summary
FINAL	Final statistics page summarizing data across the entire reporting period
INTERVAL	Statistics page for each reporting interval

If the /TABULAR qualifier is present without any keywords, then /TABULAR=(FINAL,BYCLUSTER) is assumed.

All keywords are available in negated form.

The BYCLUSTER and BYNODE keywords with /TABULAR provide two reporting possibilities for reports in cluster format when the /CLUSTER qualifier is present in the command line. In BYNODE format, memory statistics are given for each node in the cluster; in BYCLUSTER format, memory statistics are given as the average, minimum, maximum, and total values across the cluster.

The FINAL keyword causes data to be reported over the entire reporting period. For example, if the archive interval in a history file is 900 seconds (15 minutes) and you specify final statistics for a reporting period of one hour, one final statistics page is generated. The following is an example of the command:

```
$ PERFORMANCE REPORT=HISTORY/"SINCE=30-AUG-1988:10:00" -
 /BEFORE="30-AUG-1988:11:00"/TABULAR=FINAL/NOGRAPH HISTORY.DAT
```

The INTERVAL keyword with the /TABULAR qualifier causes tabular data to be reported for each interval in a reporting period, rather than over the entire reporting period itself. For each reporting interval, there is a separate statistics page.

For example, substituting INTERVAL for FINAL in the above command for the same history file will produce four interval statistics pages, one for each archive interval in the history file. Refer to the following:

```
$ PERFORMANCE REPORT=HISTORY/"SINCE=30-AUG-1988:10:00" -
 /BEFORE="30-AUG-1988:11:00"/TABULAR=INTERVAL/NOGRAPH HISTORY.DAT
```

Note that in both examples the /NOGRAPH qualifier suppresses the generation of graphs.

A reporting interval other than the archive interval is selected when you specify a daily, weekly, or monthly reporting unit as described in Section 16.5, or a user-defined reporting unit as described in Section 16.6.

16.8.3 Specifying Final and Interval Reports for Prime and Nonprime Times

The /SHIFT qualifier may be specified in a number of ways to select data for reporting. In the following examples:

/TABULAR=FINAL/NOSHIFT generates final statistics where the data from prime and nonprime time is reported.

/TABULAR=FINAL/SHIFT=PRIME produces final prime statistics.

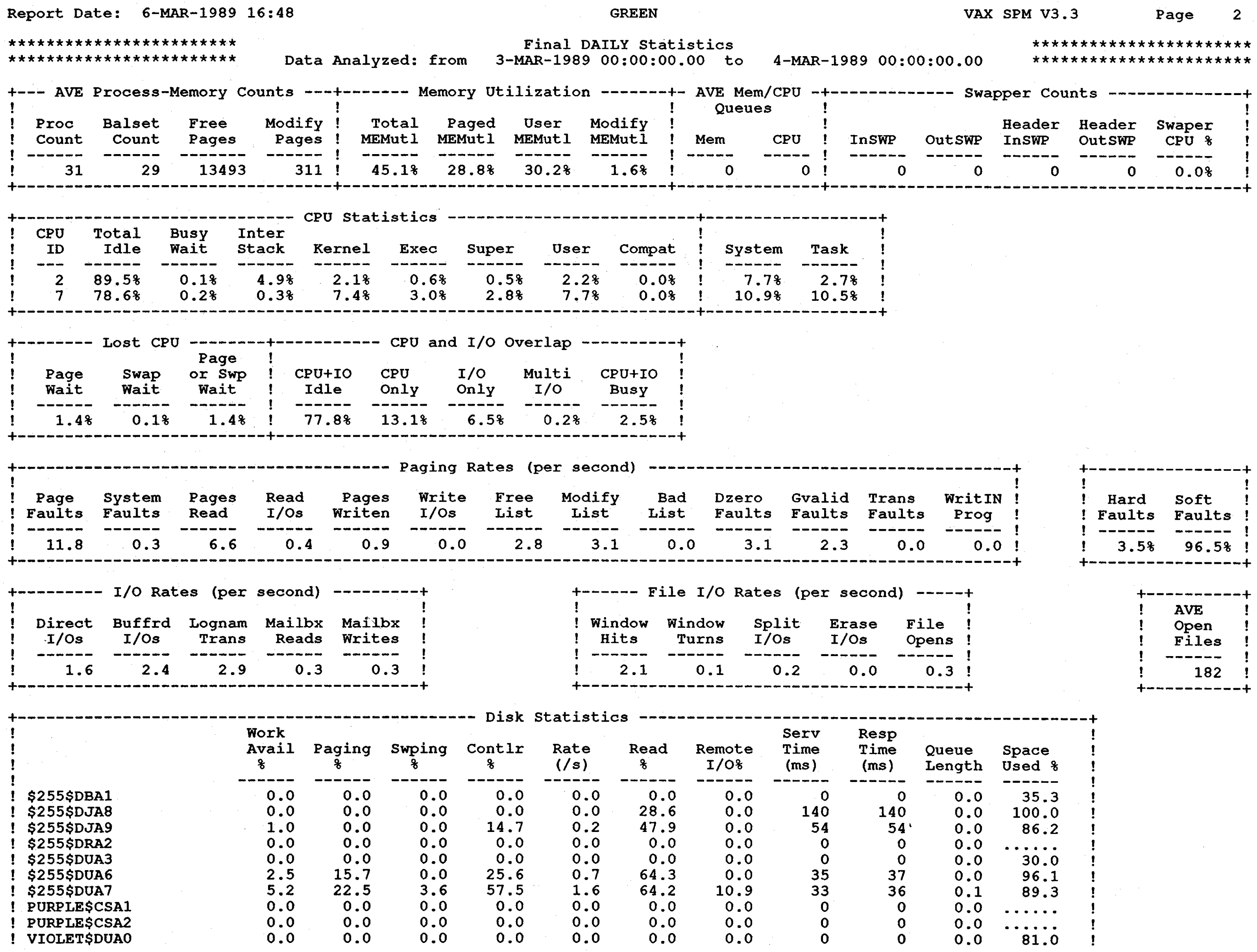
The substitution of the NONPRIME keyword for PRIME in the above examples generates nonprime statistics as appropriate.

In the following example, prime time is the period from 8 a.m. to 5 p.m. (the default). The report will contain final statistics for the entire reporting period (the period of time covered by records in HISTORY.DAT) for prime time only. Only CPU statistics are specified.

```
$ PERFORMANCE REPORT=HISTORY/CLASS=(NODEFAULT_STATISTICS,CPU) -
/TABULAR=FINAL/SHIFT=PRIME HISTORY.DAT
```

In the example below, prime time is defined as the period from 8 a.m. to 12 p.m., and from 1 p.m. to 5 p.m. The report will contain a final statistics page (default statistics only) summarized for these hours only across the period of time covered by records in HISTORY.DAT.

```
$ PERFORMANCE REPORT=HISTORY/TABULAR=FINAL/SHIFT=PRIME -
  /P_HOURS=(8-11,13-16) HISTORY.DAT
```



```
Report Date:  6-MAR-1989 16:48                     GREEN                          VAX SPM V3.3        Page    2

************************                   Final DAILY Statistics                                 ***********************
************************     Data Analyzed: from   3-MAR-1989 00:00:00.00  to   4-MAR-1989 00:00:00.00    ***********************

+--- AVE Process-Memory Counts ---+------- Memory Utilization -------+- AVE Mem/CPU -+------------- Swapper Counts --------------+
!                                 !                                   !    Queues      !                                          !
!  Proc   Balset   Free    Modify !  Total   Paged    User   Modify   !                !                  Header  Header  Swaper   !
!  Count  Count    Pages   Pages  ! MEMutl  MEMutl  MEMutl  MEMutl    !  Mem      CPU  !  InSWP  OutSWP   InSWP   OutSWP  CPU %    !
! ------  ------  -------  ------ ! ------  ------  ------  ------    ! -----   ----- !  ------  ------   ------  ------  ------   !
!    31      29    13493      311 !  45.1%   28.8%   30.2%    1.6%    !   0         0 !      0       0       0       0    0.0%   !
+---------------------------------+-----------------------------------+----------------+------------------------------------------+

+-------------------------- CPU Statistics --------------------------+-----------------+
!  CPU    Total   Busy   Inter                                        !                 !
!   ID     Idle   Wait   Stack   Kernel   Exec    Super   User  Compat !  System   Task  !
!  ---   ------  ------  ------  ------  ------  ------  ------  ------ !  ------  ------ !
!    2   89.5%    0.1%    4.9%    2.1%    0.6%    0.5%    2.2%    0.0% !   7.7%    2.7%  !
!    7   78.6%    0.2%    0.3%    7.4%    3.0%    2.8%    7.7%    0.0% !  10.9%   10.5%  !
+---------------------------------------------------------------------+-----------------+

+-------- Lost CPU --------+----------- CPU and I/O Overlap ----------+
!                   Page   !                                          !
!  Page    Swap    or Swp  !  CPU+IO   CPU     I/O    Multi   CPU+IO  !
!  Wait    Wait     Wait   !   Idle    Only    Only    I/O     Busy   !
! ------  ------   ------  !  ------  ------  ------  ------  ------  !
!   1.4%    0.1%     1.4%  !   77.8%   13.1%    6.5%    0.2%    2.5%  !
+--------------------------+------------------------------------------+

+------------------------------------- Paging Rates (per second) -------------------------------------+   +----------------+
!                                                                                                      !   !                !
!  Page   System   Pages   Read    Pages   Write    Free   Modify    Bad   Dzero   Gvalid  Trans  WritIN !   !  Hard    Soft  !
! Faults  Faults   Read    I/Os   Writen   I/Os    List    List    List   Faults  Faults  Faults  Prog  !   ! Faults  Faults !
! ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------ !   ! ------  ------ !
!   11.8     0.3     6.6     0.4     0.9     0.0     2.8     3.1     0.0     3.1     2.3     0.0    0.0 !   !   3.5%   96.5% !
+------------------------------------------------------------------------------------------------------+   +----------------+

+--------- I/O Rates (per second) ---------+               +------ File I/O Rates (per second) -----+       +----------+
!                                          !               !                                        !       !   AVE    !
!  Direct  Buffrd  Lognam  Mailbx  Mailbx  !               ! Window  Window   Split   Erase   File  !       !   Open   !
!   I/Os    I/Os    Trans   Reads  Writes  !               !  Hits    Turns   I/Os    I/Os   Opens  !       !   Files  !
!  ------  ------  ------  ------  ------  !               ! ------  ------  ------  ------  ------ !       !  ------  !
!    1.6     2.4     2.9     0.3     0.3   !               !   2.1     0.1     0.2     0.0     0.3  !       !    182   !
+------------------------------------------+               +----------------------------------------+       +----------+

+------------------------------------------------ Disk Statistics ------------------------------------------------+
!                             Work                                                        Serv    Resp                    !
!                             Avail   Paging  Swping  Contlr   Rate    Read   Remote   Time    Time    Queue   Space     !
!                              %        %       %       %      (/s)     %      I/O%    (ms)    (ms)    Length  Used %    !
!                            ------  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------    !
! $255$DBA1                    0.0     0.0     0.0     0.0     0.0     0.0     0.0       0       0     0.0    35.3    !
! $255$DJA8                    0.0     0.0     0.0     0.0     0.0    28.6     0.0     140     140     0.0   100.0    !
! $255$DJA9                    1.0     0.0     0.0    14.7     0.2    47.9     0.0      54      54     0.0    86.2    !
! $255$DRA2                    0.0     0.0     0.0     0.0     0.0     0.0     0.0       0       0     0.0  ......    !
! $255$DUA3                    0.0     0.0     0.0     0.0     0.0     0.0     0.0       0       0     0.0    30.0    !
! $255$DUA6                    2.5    15.7     0.0    25.6     0.7    64.3     0.0      35      37     0.0    96.1    !
! $255$DUA7                    5.2    22.5     3.6    57.5     1.6    64.2    10.9      33      36     0.1    89.3    !
! PURPLE$CSA1                  0.0     0.0     0.0     0.0     0.0     0.0     0.0       0       0     0.0  ......    !
! PURPLE$CSA2                  0.0     0.0     0.0     0.0     0.0     0.0     0.0       0       0     0.0  ......    !
! VIOLET$DUA0                  0.0     0.0     0.0     0.0     0.0     0.0     0.0       0       0     0.0    81.0    !
```

Figure 16–6 Final Statistics Report Example

Note that the INTERVAL keyword with the /TABULAR qualifier could be used with /SHIFT=PRIME to produce Interval Statistics reports for prime hours. Likewise, using /TABULAR=INTERVAL and the /NOSHIFT qualifiers together could produce interval statistics where data from prime and nonprime time is averaged together. Table 16–5 shows the results of /TABULAR=INTERVAL for daily, weekly and monthly reporting units. Since history files can contain a lot of intervals, it may be more efficient to specify a period of interest using the /BEFORE and /SINCE qualifiers as described in Section 16.2.

16.9 Additional Graphing Features

Use the /GRAPH qualifier with the PERFORMANCE REPORT=HISTORY command to generate graphs of system metric data.

Read the *VAX SPM Reference Manual* for a full description of graphs and their contents.

In addition to using the features described in Section 16.1 to Section 16.7, you can:

- Generate graphs for a variety of system metrics.
- Generate working graphs in ANSI format, or presentation graphs in ReGIS or sixel format.
- Specify the time intervals to graph.
- Specify threshold values for graph statistics.
- Selectively inhibit graph generation based upon threshold values you define.
- Specify the a maximum scaling value to override the autoscale feature.

16.9.1 Specifying the System Metrics to Graph

The /GRAPH qualifier specifies that a hard copy graph is desired, and names the types of items to be graphed. If /GRAPH is omitted, or specified without a graph item, a graph showing CPU utilization is produced. If /GRAPH is specified, any of the graph items shown in Table 16–8 can be used as graph item keywords. If you specify the keyword ALL, you can exclude selected graphs by also specifying their keywords prefixed by NO. Certain graphs are not available when the /CLUSTER qualifier is used with the REPORT=HISTORY command (cluster format reporting); these keywords are identified by a superscript 1.

Table 16–8 Keywords for the /GRAPH Qualifier

Keyword	Meaning
ALL	All optional graphs
BALANCE_SET	Average number of processes in balance set

Table 16–8 (Cont.) Keywords for the /GRAPH Qualifier

Keyword	Meaning
BUFFERED_IO	Average number of buffered I/O requests
CPU	Percentage of time the CPU was busy
DEVICE[1]	Device I/O rate per second
DIRECT_IO	Average number of direct I/O requests
D_ALLOCATION[1]	Percentage of disk volume storage allocated
D_RATE	Average number of I/O requests per second
D_RESPONSE	Average disk response time in milliseconds
D_UTILIZATION[1]	Percentage of time the disk was busy
FILE_OPENS	Average number of file open requests
LOGICAL_NAME	Average number of logical name translation requests
LOST_TIME	Percentage of time CPU waited for paging and/or swapping I/O to complete
MAILBOX_READS	Average number of mailbox read I/O requests
MEMORY	Percentage of memory utilized
OPEN_FILES	Average number of files open at the same time
PAGE_FAULTS	Average page fault rate
PROCESS_COUNT	Average number of processes in the system
Q_CPU	Average length of the CPU queue
Q_MEMORY	Average length of the memory queue
SUMMARY[1]	Percentage graph of the CPU+IO overlap
SWAP_COUNT	Average number of swapped processes
WINDOW_HIT[1]	Percent ratio of window hits to window turns

[1] Graph not available when /CLUSTER is present in REPORT=HISTORY command line.

The command below generates a working style graph in ANSI format showing the average number of processes in the balance set. The time period for data reporting as specified by the /SINCE and /BEFORE qualifiers is one day; therefore, the report will contain one graph. Note that if the number of intervals in a reporting period exceeds 105, the graph is printed on additional pages. The /NOTABULAR qualifier suppresses generation of tabular reports.

```
$ PERFORMANCE REPORT=HISTORY/SINCE="02-JUL-1986 0" -
  /BEFORE="03-JUL-1986 0"/GRAPH=(BALANCE_SET) -
  /NOTABULAR HISTORY.DAT
```

The command below generates working style graphs showing CPU utilization and Direct I/O requests plotted against time of day. There is one plotted value per archive interval (default time unit).

```
$ PERFORMANCE REPORT=HISTORY/GRAPH=(CPU,DIRECT_IO) -
 /NOTABULAR  HISTORY.DAT
```

Several different types of graphs are possible, depending on the data to be plotted. System metrics values are plotted along the y axis either as percentages (0%–100%), or by using an autoscale that adjusts itself to the actual values present. Section 16.9.5 describes the autoscale feature and how to override it. Plotting characters can be simple or compound. With simple plotting characteristics, a column of asterisks is used. With compound plotting characteristics, y axis values are shown as comprised of several different parts using different characters to produce a "column" value. For example, CPU utilization may show a graph column made up of the characters U (user), K (kernel), and I (interrupt stack). Time intervals are plotted along the x axis.

Figures 16–7 and 16–8 show examples of graphs generated for CPU utilization (% y axis, compound plotting) and Buffered I/O statistics (autoscale y axis, simple plotting).

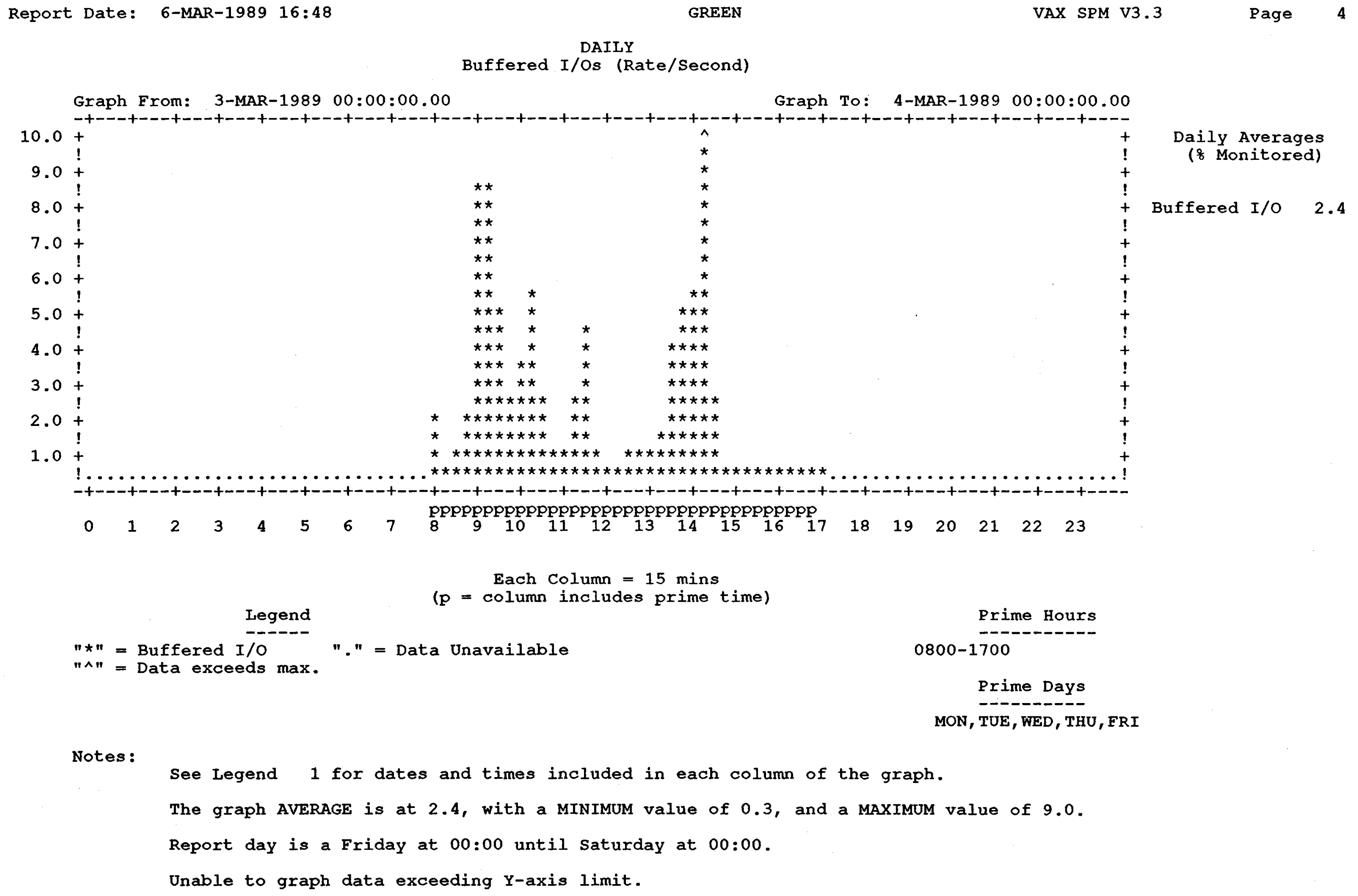
16.9.2 Specifying the Style of Graphs

Use the /STYLE qualifier with one of the four keywords shown in Table 16–9 to specify one of four different styles of graph.

Figure 16–7 CPU Utilization Report Graph

```
Report Date:  6-MAR-1989 16:48                                 GREEN                             VAX SPM V3.3      Page    5
                                                     DAILY
                                           CPU Utilization (Percent)

      Graph From:  3-MAR-1989 00:00:00.00                              Graph To:  4-MAR-1989 00:00:00.00
      -+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+----
 100% +                                                                                                +     Daily Averages
      !                                                                                                !      (% Monitored)
  90% +                                                                                                +
      !                                                                                                !
  80% +                                                                                                +   Interrupt     2.6%
      !                                                                                                !   Busy Wait     0.1%
  70% +                                                                                                +   Kernel        4.7%
      !                                                                                                !   Exec          1.8%
  60% +                                    U                                                           +   Super         1.6%
      !                                    UU                                                          !   User          4.9%
  50% +                                    UU                  U                                       +   Compat        0.0%
      !                                    SU                  S                                       !
  40% +                                    EUU                 SU                                      +
      !                                    EUU                USU                                      !
  30% +                                U   KEU U              SSS                                      +
      !                                U   KKS U             USEE                                      !
  20% +                                E   KKE UU            UEEK                                      +
      !                                K   KKKUKU   UU      UUKKK                                      !
  10% +                                K  UKKKKKKK  KK     UKKKKKU                                     +
      !................................IUUKIIIIIIII II     KIIIIIK   UU     ...........................!
      -+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+----
                                       pppppppppppppppppppppppppppppppppppp
       0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23

                                             Each Column = 15 mins
                                        (p = column includes prime time)
                      Legend                                                                     Prime Hours
                      ------                                                                     -----------
      "I" = Interrupt         "B" = Busy Wait                                              0800-1700
      "K" = Kernel            "E" = Exec
      "S" = Super             "U" = User
      "C" = Compat            "." = Data Unavailable

                                                                                           Prime Days
                                                                                           ----------
                                                                                       MON,TUE,WED,THU,FRI

      Notes:
               See Legend   1 for dates and times included in each column of the graph.

               The graph AVERAGE is at 15.7%, with a MINIMUM value of 2.2%, and a MAXIMUM value of 58.4%.

               Report day is a Friday at 00:00 until Saturday at 00:00.
```

Figure 16–8 Buffered I/O Statistics Report Graph



```
Report Date:  6-MAR-1989 16:48                                 GREEN                           VAX SPM V3.3        Page    4
                                                     DAILY
                                          Buffered I/Os (Rate/Second)

      Graph From:  3-MAR-1989 00:00:00.00                              Graph To:  4-MAR-1989 00:00:00.00
      -+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+----
 10.0 +                                                         ^                                      +   Daily Averages
      !                                                         *                                      !     (% Monitored)
  9.0 +                                                         *                                      +
      !                                    **                   *                                      !
  8.0 +                                    **                   *                                      +  Buffered I/O   2.4
      !                                    **                   *                                      !
  7.0 +                                    **                   *                                      +
      !                                    **                   *                                      !
  6.0 +                                    **                   *                                      +
      !                                    **   *              **                                      !
  5.0 +                                    ***  *             ***                                      +
      !                                    ***  *    *        ***                                      !
  4.0 +                                    ***  *    *       ****                                      +
      !                                    *** **    *       ****                                      !
  3.0 +                                    *** **    *       ****                                      +
      !                                    *******  **       *****                                     !
  2.0 +                                *  ********  **       *****                                     +
      !                                *  ********  **      ******                                     !
  1.0 +                                * **************  *********                                     +
      !................................*************************************...........................!
      -+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+----
                                       pppppppppppppppppppppppppppppppppppp
       0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23

                                             Each Column = 15 mins
                                       (p = column includes prime time)
                      Legend                                                            Prime Hours
                      ------                                                            -----------
      "*" = Buffered I/O      "." = Data Unavailable                                0800-1700
      "^" = Data exceeds max.
                                                                                        Prime Days
                                                                                        ----------
                                                                                   MON,TUE,WED,THU,FRI

      Notes:
               See Legend   1 for dates and times included in each column of the graph.

               The graph AVERAGE is at 2.4, with a MINIMUM value of 0.3, and a MAXIMUM value of 9.0.

               Report day is a Friday at 00:00 until Saturday at 00:00.

               Unable to graph data exceeding Y-axis limit.
```

Table 16–9 Keywords for the /STYLE Qualifier

Keyword	Meaning
TOTAL	Working style graph in ANSI format showing the total value for each graph interval
RANGE	Working style graph in ANSI showing the minimum, average, and maximum values for each graph interval
SIXEL	Presentation graphs produced in sixel output format
REGIS	Presentation graphs produced in ReGIS output format

16.9.2.1 Specifying Working Style Graphs in ANSI

Working graphs are stacked bar graphs in ANSI format. They can be printed and displayed using conventional printers and terminals. VAX SPM produces two types of working graphs: Total and Range, specified by using the TOTAL and RANGE keywords with the /STYLE qualifier.

The Total working graph produces a stacked bar graph in ANSI format showing the total of one metric over time. Using the REPORT=HISTORY/CLUSTER command, the cluster total for that metric is shown.

The Range type of working graph is used with the REPORT=HISTORY/CLUSTER command for cluster analysis. It graphs the minimum for a node, the maximum for a node, and the cluster average.

16.9.2.2 Specifying Presentation Graphs

Presentation graphs are graphs of performance data suitable for inclusion in management reports and slide presentations. These graphs can be produced in either sixel or ReGIS format.

Generate presentation graphs in sixel format by specifying /STYLE=SIXEL, and in ReGIS format by specifying /STYLE=REGIS.

The following command produces a default version of a presentation quality graph of CPU utilization in ReGIS format:

```
$ PERFORMANCE REPORT=HISTORY/GRAPH/STYLE=REGIS
```

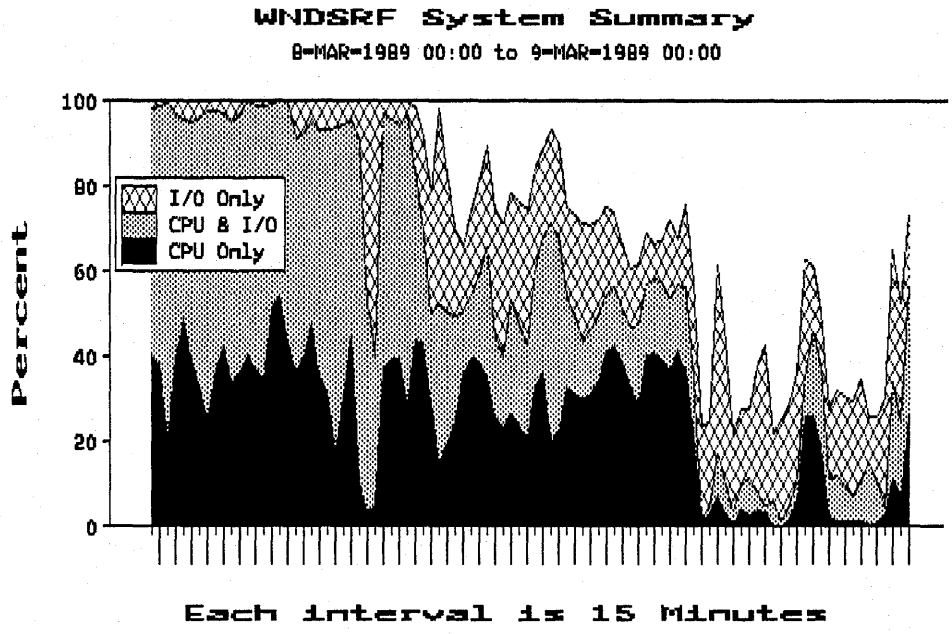
The following are characteristics of default presentation graphs:

- Graph type is a filled line graph.
- A legend is supplied, which maps colors or line types to the metrics being graphed.
- The title is supplied, which contains the following:
  - Node or cluster name
  - Disk or device identifier (if disk and device metrics are graphed)
  - The graph metric name, such as CPU, Memory, or Page Faults
- The subtitle is supplied, which describes the graph reporting period.
- The horizontal label is supplied, which describes the reporting interval.

- Colors from palette number 1 are generated.
- The vertical label is supplied with the unit appropriate to the metric being graphed. For example, the vertical label for a CPU graph is percent (%), for a D_RATE it is rate/second, and BALANCE_SET is a number representing the number of processes.

Figure 16–9 is an example of a presentation graph.

Figure 16–9 Default Presentation Graph



A file is created for each graph according to naming conventions described in Chapter 17. The graph generated by the above command is on node GREEN in a file named the following:

DISK$USER_A:SPM$GRAH_CPU_1.REGIS_GREEN

In addition to selecting ReGIS and sixel format, you can choose from four types of presentation graphs:

- Line
- Histogram
- Pie chart
- Stacked bar

The graph types can be produced using a variety of style characteristics such as color, borders, grids, and customized titles and labels. Graph types and characteristics other than the default are specified using a graph description file, which is described in Chapter 17. They can be printed and displayed using ReGIS-compatible output devices.

16.9.2.3 Requirements

To print presentation graphs in either ReGIS or sixel format requires ReGIS-compatible output devices such as an LN03, LCG01, LA34_VA, or LA50.

It is possible to produce sixel files from VT240 and VT340 color terminals for printing on monochrome devices such as the LN03. To do this, set the COLOR PRINTING set-up attribute on the terminal to MONOCHROME. Failure to do this produces a solid black rectangle in place of a graph.

To display presentation graphs requires a ReGIS-compatible terminal such as the VT125, VT240, VT241, VT330, VT340, or PC350.

An important difference between the ReGIS and sixel graphs is the manner in which they are produced. Graphs in ReGIS format are not displayed as they are produced; therefore, they may be produced on any type of terminal. Graphs in sixel format are generated by a screen dump method in which each graph is displayed on the screen as it is produced. Therefore, sixel graphs must be generated on a ReGIS-compatible terminal such as the VT125, VT240, VT241, VT330, VT340, or PC350.

Sixel graph files can be included in VAX DOCUMENT source files.

16.9.3 Specifying Time Intervals for Plotting

By default, the time unit for plotting data is the the archive interval for the history file.

With the REPORT=HISTORY command, the default plotting unit is changed when you specify daily, weekly, or monthly reporting units as described in Section 16.5 or user-defined reporting units as described in Section 16.6.

There are a maximum of 105 units per graph; if necessary, individual graphs continue on additional pages.

16.9.4 Selective Generation of Graphs

The ability to selectively generate graphs lets you describe performance criteria to determine when graphs of disk and device statistics will be produced. For example, if you have 50 disks on your system and you are only interested in the I/O rates for the busiest disks, you can specify that graphs be generated only for the disks that have more than 10 I/Os per second over 30% of the time.

Specifying the selective generation of graphs requires two steps:

1 Specify threshold values for graph items, devices, and disks.

2 Use the /SELECT qualifier to test the threshold values and determine whether or not to generate graphs.

Specifying the selective generation of graphs in the SPM$MANAGER.COM command procedure (Chapter 3) lets VAX SPM automatically monitor your system and generate graphs for the disks and devices on which resource utilization exceeds the thresholds you define.

16.9.4.1 Specifying Thresholds for Graph Items, Devices and Disks

A threshold is a user-defined value associated with a graph. Thresholds work in conjunction with the /SELECT qualifier to determine if a graph is generated for a graph item. When the item is graphed using a threshold value, the threshold value is represented as a series of dotted lines on working graphs.

Each graph item specified using /GRAPH can be accompanied by a threshold value. The threshold value must be a positive integer. In addition, the /DEVICE and /DISK qualifiers allow threshold values to be specified locally for each device or disk; these values override the global threshold values given with /GRAPH and its keywords.

In other words, a threshold value given with the /GRAPH qualifier can be overridden by a threshold value given for a specific device by using the /DEVICE qualifier. Likewise, a value given to a D_ keyword can be overridden by a threshold value given for a specific disk using the /DISK qualifier.

Consider how global and local threshold values are specified by the following commands:

```
$ PERFORMANCE REPORT=HISTORY/DEVICE=(XEA0:30,XEA1)/CLASS=DEVICE -
  /GRAPH=(DEVICE:20)/NOTABULAR HISTORY.DAT
```

In the above command:

- /DEVICE=(XEA0:30,XEA1), specifies a threshold of 30 for device XEA0, which overrides the global threshold of 20 specified by /GRAPH=(DEVICE:20).
- Since no threshold is given for device XEA1 by /DEVICE=(XEA0:30,XEA1), the threshold of 20 specified globally by /GRAPH=(DEVICE:20) is used.
- /CLASS=DEVICE is included since device statistics are not among the default classes for reporting.
- /GRAPH with the DEVICE keyword declares that device I/O rate graphs will be generated for the devices specified by the /DEVICE qualifier.

The following command specifies threshold values for disk allocation graphs:

```
$ PERFORMANCE REPORT=HISTORY/GRAPH=(D_ALLOCATION:70) -
   /DISK=(DBA0:60,DBA1)/NOTABULAR HISTORY.DAT
```

In the above command:

- /GRAPH=(D_ALLOCATION:70) assigns a global threshold fo 70 for all D_ALLOCATION items; therefore, DBA1 has a threshold value of 70.
- /DISK=(DBA0:60, DBA1) specifies that graphs will be generated for disks DBA0 and DBA1. The locally assigned threshold value of 60 for disk DBA0 overrides the globally assigned value of 70 given by /GRAPH.

Note that in contrast to specifying disks for data collection, disk names may not contain leading underscores. If /DISK is omitted, then /DISK=ALL (for all disks in the history file) is assumed.

16.9.4.2 Using the /SELECT Qualifier

The /SELECT qualifier has two keywords, shown in Table 16–10.

Table 16–10 Keywords for the /SELECT Qualifier

Keyword	Meaning
GREATER_ THAN:percent	At least "percent" of the graph columns must be greater than or equal to the threshold value specified with the /GRAPH=, /DISK, or /DEVICE qualifiers.
LESS_THAN:percent	At least "percent" of the graph columns must be less than the threshold value specified with the /GRAPH=, /DISK, or /DEVICE qualifiers.

If the keyword GREATER_THAN is specified with /SELECT, then to generate a graph, at least "percent" of the graph columns must be greater than or equal to the threshold value specified with the /GRAPH=, /DISK, or /DEVICE qualifiers. If the keyword LESS_THAN is specified with /SELECT, at least "percent" of the graph columns must be less than or equal to the threshold value specified with the /GRAPH=, /DISK, or /DEVICE qualifiers.

The GREATER_THAN and LESS_THAN keywords themselves accept a single value representing the percentage of columns graphed that must meet the threshold criteria before the graph is produced. If the GREATER_THAN keyword is specified without a value, then 30 percent is assumed. If the LESS_THAN keyword is specified without a value, then 90 percent is assumed.

If the /SELECT qualifier is present without a keyword, then GREATER_THAN:50 is assumed. If 0 is given as the percent, then only one point on the graph has to satisfy the criterion to produce the graph.

To specify that disk rate graphs be generated only for disks that have more than 10 I/Os per second over 30% of the time, you would use a command similar to the following:

```
/GRAPH=D_RATE:10/SELECT=GREATER_THAN:30
```

16.9.5 Specifying a Scaling Factor to Override Autoscaling

VAX SPM automatically establishes the scale for the vertical axis of graphs based upon the range of the units graphed. For example, if the number of page faults per seconds is in the range of 0 to 160, increments on the vertical axis in units of 16 are labeled 16, 32, 64, ... to 160. If the number of page faults graphed is in the range of 0 to 60, the increments on the graph are in units of 6 are labeled 6, 12, 24, ... to 60.

You can override the autoscaling mechanism by specifying the number you want at the top of the vertical axis as a second value to the /GRAPH, /DISK, or /DEVICE qualifier. If you specify a maximum scaling value that is smaller than the actual value of the display-variable, the graph displays a carat (^) as the top line of the bar to indicate that the actual value is greater than the graph allows.

The following example specifies a threshold of 100 and a maximum scale value of 150:

```
/GRAPH=PAGE:100:150
```

If you wish to specify a maximum scale value of 60 and no threshold value, specify a threshold of 0 as in the following example:

```
/GRAPH=CPU:0:60
```

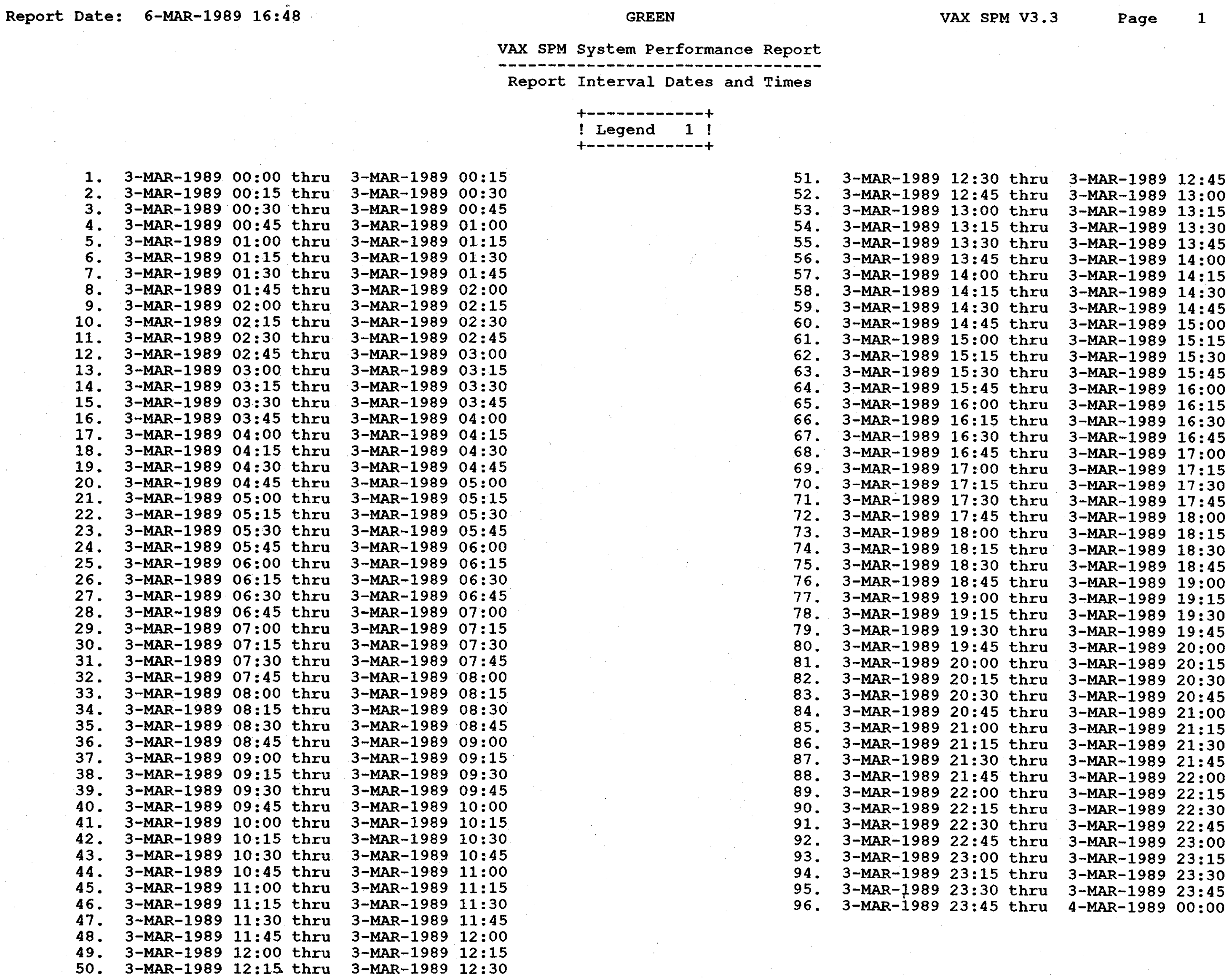
The scaling on stacked bar graphs is predetermined and therefore is not affected by scaling values specified for the /DISK, /DEVICE, and /GRAPH qualifiers.

16.9.6 Specifying a Legend for Graphs

The /LEGEND qualifier causes a legend page to be generated for each set of output statistics in tabular or graph form, detailing what periods of time were included in each interval of the report. For graph reports, the legend shows the time periods over which data was averaged to produce each point or column on the graph. For tabular reports consisting of interval statistics, there is a report page for each time period given by the legend page. Tabular values are for these time periods. Figure 16–10 shows an example of a legend page.

The Legend page reproduces in different format the contents of the dates file (Section 16.5 and Section 16.6.1). On the graph page, numbers along the x axis correspond to the numbered intervals on the Legend page. The Notes on a graph page direct the user to the corresponding Legend or Report Interval page.

Figure 16–10 Legend Page of a Graph



```
Report Date:  6-MAR-1989 16:48                        GREEN                        VAX SPM V3.3      Page    1

                                          VAX SPM System Performance Report
                                          ---------------------------------
                                           Report Interval Dates and Times

                                                   +-------------+
                                                   ! Legend   1 !
                                                   +-------------+

       1.  3-MAR-1989 00:00 thru  3-MAR-1989 00:15                        51.  3-MAR-1989 12:30 thru  3-MAR-1989 12:45
       2.  3-MAR-1989 00:15 thru  3-MAR-1989 00:30                        52.  3-MAR-1989 12:45 thru  3-MAR-1989 13:00
       3.  3-MAR-1989 00:30 thru  3-MAR-1989 00:45                        53.  3-MAR-1989 13:00 thru  3-MAR-1989 13:15
       4.  3-MAR-1989 00:45 thru  3-MAR-1989 01:00                        54.  3-MAR-1989 13:15 thru  3-MAR-1989 13:30
       5.  3-MAR-1989 01:00 thru  3-MAR-1989 01:15                        55.  3-MAR-1989 13:30 thru  3-MAR-1989 13:45
       6.  3-MAR-1989 01:15 thru  3-MAR-1989 01:30                        56.  3-MAR-1989 13:45 thru  3-MAR-1989 14:00
       7.  3-MAR-1989 01:30 thru  3-MAR-1989 01:45                        57.  3-MAR-1989 14:00 thru  3-MAR-1989 14:15
       8.  3-MAR-1989 01:45 thru  3-MAR-1989 02:00                        58.  3-MAR-1989 14:15 thru  3-MAR-1989 14:30
       9.  3-MAR-1989 02:00 thru  3-MAR-1989 02:15                        59.  3-MAR-1989 14:30 thru  3-MAR-1989 14:45
      10.  3-MAR-1989 02:15 thru  3-MAR-1989 02:30                        60.  3-MAR-1989 14:45 thru  3-MAR-1989 15:00
      11.  3-MAR-1989 02:30 thru  3-MAR-1989 02:45                        61.  3-MAR-1989 15:00 thru  3-MAR-1989 15:15
      12.  3-MAR-1989 02:45 thru  3-MAR-1989 03:00                        62.  3-MAR-1989 15:15 thru  3-MAR-1989 15:30
      13.  3-MAR-1989 03:00 thru  3-MAR-1989 03:15                        63.  3-MAR-1989 15:30 thru  3-MAR-1989 15:45
      14.  3-MAR-1989 03:15 thru  3-MAR-1989 03:30                        64.  3-MAR-1989 15:45 thru  3-MAR-1989 16:00
      15.  3-MAR-1989 03:30 thru  3-MAR-1989 03:45                        65.  3-MAR-1989 16:00 thru  3-MAR-1989 16:15
      16.  3-MAR-1989 03:45 thru  3-MAR-1989 04:00                        66.  3-MAR-1989 16:15 thru  3-MAR-1989 16:30
      17.  3-MAR-1989 04:00 thru  3-MAR-1989 04:15                        67.  3-MAR-1989 16:30 thru  3-MAR-1989 16:45
      18.  3-MAR-1989 04:15 thru  3-MAR-1989 04:30                        68.  3-MAR-1989 16:45 thru  3-MAR-1989 17:00
      19.  3-MAR-1989 04:30 thru  3-MAR-1989 04:45                        69.  3-MAR-1989 17:00 thru  3-MAR-1989 17:15
      20.  3-MAR-1989 04:45 thru  3-MAR-1989 05:00                        70.  3-MAR-1989 17:15 thru  3-MAR-1989 17:30
      21.  3-MAR-1989 05:00 thru  3-MAR-1989 05:15                        71.  3-MAR-1989 17:30 thru  3-MAR-1989 17:45
      22.  3-MAR-1989 05:15 thru  3-MAR-1989 05:30                        72.  3-MAR-1989 17:45 thru  3-MAR-1989 18:00
      23.  3-MAR-1989 05:30 thru  3-MAR-1989 05:45                        73.  3-MAR-1989 18:00 thru  3-MAR-1989 18:15
      24.  3-MAR-1989 05:45 thru  3-MAR-1989 06:00                        74.  3-MAR-1989 18:15 thru  3-MAR-1989 18:30
      25.  3-MAR-1989 06:00 thru  3-MAR-1989 06:15                        75.  3-MAR-1989 18:30 thru  3-MAR-1989 18:45
      26.  3-MAR-1989 06:15 thru  3-MAR-1989 06:30                        76.  3-MAR-1989 18:45 thru  3-MAR-1989 19:00
      27.  3-MAR-1989 06:30 thru  3-MAR-1989 06:45                        77.  3-MAR-1989 19:00 thru  3-MAR-1989 19:15
      28.  3-MAR-1989 06:45 thru  3-MAR-1989 07:00                        78.  3-MAR-1989 19:15 thru  3-MAR-1989 19:30
      29.  3-MAR-1989 07:00 thru  3-MAR-1989 07:15                        79.  3-MAR-1989 19:30 thru  3-MAR-1989 19:45
      30.  3-MAR-1989 07:15 thru  3-MAR-1989 07:30                        80.  3-MAR-1989 19:45 thru  3-MAR-1989 20:00
      31.  3-MAR-1989 07:30 thru  3-MAR-1989 07:45                        81.  3-MAR-1989 20:00 thru  3-MAR-1989 20:15
      32.  3-MAR-1989 07:45 thru  3-MAR-1989 08:00                        82.  3-MAR-1989 20:15 thru  3-MAR-1989 20:30
      33.  3-MAR-1989 08:00 thru  3-MAR-1989 08:15                        83.  3-MAR-1989 20:30 thru  3-MAR-1989 20:45
      34.  3-MAR-1989 08:15 thru  3-MAR-1989 08:30                        84.  3-MAR-1989 20:45 thru  3-MAR-1989 21:00
      35.  3-MAR-1989 08:30 thru  3-MAR-1989 08:45                        85.  3-MAR-1989 21:00 thru  3-MAR-1989 21:15
      36.  3-MAR-1989 08:45 thru  3-MAR-1989 09:00                        86.  3-MAR-1989 21:15 thru  3-MAR-1989 21:30
      37.  3-MAR-1989 09:00 thru  3-MAR-1989 09:15                        87.  3-MAR-1989 21:30 thru  3-MAR-1989 21:45
      38.  3-MAR-1989 09:15 thru  3-MAR-1989 09:30                        88.  3-MAR-1989 21:45 thru  3-MAR-1989 22:00
      39.  3-MAR-1989 09:30 thru  3-MAR-1989 09:45                        89.  3-MAR-1989 22:00 thru  3-MAR-1989 22:15
      40.  3-MAR-1989 09:45 thru  3-MAR-1989 10:00                        90.  3-MAR-1989 22:15 thru  3-MAR-1989 22:30
      41.  3-MAR-1989 10:00 thru  3-MAR-1989 10:15                        91.  3-MAR-1989 22:30 thru  3-MAR-1989 22:45
      42.  3-MAR-1989 10:15 thru  3-MAR-1989 10:30                        92.  3-MAR-1989 22:45 thru  3-MAR-1989 23:00
      43.  3-MAR-1989 10:30 thru  3-MAR-1989 10:45                        93.  3-MAR-1989 23:00 thru  3-MAR-1989 23:15
      44.  3-MAR-1989 10:45 thru  3-MAR-1989 11:00                        94.  3-MAR-1989 23:15 thru  3-MAR-1989 23:30
      45.  3-MAR-1989 11:00 thru  3-MAR-1989 11:15                        95.  3-MAR-1989 23:30 thru  3-MAR-1989 23:45
      46.  3-MAR-1989 11:15 thru  3-MAR-1989 11:30                        96.  3-MAR-1989 23:45 thru  4-MAR-1989 00:00
      47.  3-MAR-1989 11:30 thru  3-MAR-1989 11:45
      48.  3-MAR-1989 11:45 thru  3-MAR-1989 12:00
      49.  3-MAR-1989 12:00 thru  3-MAR-1989 12:15
      50.  3-MAR-1989 12:15 thru  3-MAR-1989 12:30
```

16.10 Dumping Data Used to Generate Reports and Graphs

The contents of the data records used to generate the tabular and graph pages can be dumped in ASCII or binary format using the /DUMP qualifier. The dump file is an RMS sequential file of variable length records. The records themselves contain "computed" data, and are grouped per interval over the reporting period requested in the report or graph.

By default, the dump file is in binary (NOASCII) format and has the name SPMDUMP.DAT. Using the FILE and TYPE keywords, you can specify the dump file name and change the format to ASCII.

The /DUMP qualifier is provided for those users who want to write their own programs for reporting on performance data, or who wish to use a graphics package such as DATATRIEVE or DECgraph to format data and produce reports. DATATRIEVE record definition files are provided in the example directory and also in Appendix A. SPMASCII.DTR provides a definition for an ASCII formatted file, and SPMBINARY.DTR provides a definition for a binary (NOASCII) formatted file.

DATATRIEVE definitions may be found on-line in the VAX SPM example in SPM$EXAMPLES:.

In the example below, the history file HISTORY.DAT is dumped in binary format to the file HIST1.DMP. Format is that for a single node, and the node is PURPLE.

```
$ PERFORMANCE REPORT=HISTORY/NODE=PURPLE -
  /DUMP=(FILE=HIST1.DMP,TYPE=NOASCII) HISTORY.DAT
```

16.11 Reporting Cluster Information

The cluster support qualifiers /CLUSTER and /NODE allow two formats for reporting:

- Reporting of data for a single node, in single-node format
- Reporting of data for multiple nodes, in cluster format

The presence of /CLUSTER in the command line indicates that a cluster analysis of the data from 1 to n nodes is to be performed. When cluster analysis is indicated, the format of the resulting tabular report, if selected, will be different from that of a single-node analysis, and the number of different graphs available will be less than those for single-node analysis. In addition, if the type of the graph depicts a “total” value such as direct I/O rate per second, the “total” will be the total of the data from all nodes being analyzed as opposed to the total from any single node, as in the case of single-node analysis.

The /CLUSTER qualifier has the following format:

```
/CLUSTER(=[NAME=cluster-name])
```

The optional NAME keyword with /CLUSTER allows you to specify a cluster name, which will be printed at the top of each report page.

The /NODE qualifier allows you to select the specific node or nodes for which the analysis is to be performed. If there are no arguments supplied to the /NODE qualifier, then data for all nodes found in the input file will be analyzed as a single VAXcluster system.

The following is an example of the /NODE qualifier format:

```
/NODE=([node,...])
```

To report data for a single node in single-node form, the command line is:

```
$ PERFORMANCE REPORT=HISTORY/NODE=node-name...
```

To report data for selected multiple nodes in cluster form, the command line is:

```
$ PERFORMANCE REPORT=HISTORY/CLUSTER-
  /NODE=(node-name,...)
```

If the /NODE qualifier is omitted from the command line in either of the above examples, data for all nodes found in the database is included in the Cluster report.

17 Generating Presentation Graphs

Chapter 14 and Chapter 16 described how to generate default presentation quality graphs by using the REPORT=LOG_FILE and the REPORT=HISTORY commands. This chapter describes how to generate nondefault presentation quality graphs, and provides the following information:

- Description of presentation quality graphs
- Requirements for generating presentation quality graphs
- Commands for generating presentation quality graphs
- Graph description file and how to use it
- Graph file-naming conventions
- Graph selection recommendations
- Examples of generating histogram, stacked bar, and pie chart graphs

17.1 Description of Presentation Quality Graphs

Presentation quality graphs are graphs of performance metrics in ReGIS and sixel format, which can be included in management reports and slide presentations.

There are four types of presentation quality graphs:

- Line
- Histogram
- Stacked bar
- Pie chart

These graphs can be generated in ReGIS or sixel format with characteristics such as color, borders, grids, and customized titles and labels.

17.2 Requirements

To print presentation graphs in either ReGIS or sixel format requires ReGIS-compatible output devices such as an LN03R, LCG01, LA34_VA, or LA50.

It is possible to produce sixel files from VT240 and VT340 color terminals for printing on monochrome devices such as the LN03. To do this, set the COLOR PRINTING set-up attribute on the terminal to MONOCHROME. Failure to do this produces a solid black rectangle in place of a graph.

To display presentation quality graphs requires a ReGIS-compatible terminal such as the VT125, VT240, VT241, VT330, VT340, or PC350.

An important difference between the ReGIS and sixel graphs is the manner in which they are produced. Graphs in ReGIS format are not displayed as they are produced; therefore, they may be produced on any type of a terminal. Graphs in sixel format are generated by a screen dump method in which each graph is displayed on the screen as it is produced. Therefore, sixel graphs must be generated on a ReGIS-compatible terminal such as the VT125, VT240, VT241, VT330, VT340, or PC350.

Sixel graph files can be included in VAX DOCUMENT source files.

17.3 Commands

To generate presentation quality graphs with the REPORT=LOG_FILE or REPORT=HISTORY commands:

- Specify the /GRAPH qualifier and a graph item such as CPU, memory, or page faults.

  Section 14.7.1 describes specifying graph items for the REPORT=LOG_FILE command and Section 16.9.1 describes specifying graphs items for the REPORT=HISTORY command.

- Specify the /STYLE qualifier and a graph format:
  - Specify /STYLE=REGIS for graphs in ReGIS format.
  - Specify /STYLE=SIXEL for graphs in sixel format.

The following command generates a filled line graph (the default type) presentation quality graph in ReGIS format from data in a log file:

```
$ PERFORMANCE REPORT=LOG_FILE/GRAPH(=BALANCE_SET) -
  /STYLE=REGIS SPM$COLLECT_TUNE.DAT
```

The command below generates default type presentation quality graph in sixel format from data in a history file:

```
$ PERFORMANCE REPORT=HISTORY/GRAPH=BALANCE_SET/STYLE=SIXEL HISTORY.DAT
```

It may take several seconds for sixel graphs to be written to a file. During this time, do not type any commands because commands typed from the keyboard can corrupt the graph.

Default presentation quality graphs have the following characteristics:

- Graph type is a filled line graph.
- A legend is supplied, which maps colors or line types to the metrics being graphed.
- The title is supplied, which contains the following:
  - Node or cluster name
  - Disk or device identifier (if disk and device metrics are graphed)
  - The graph metric name, such as CPU, Memory, or Page_Faults
- The subtitle is supplied, which describes the graph reporting period.
- The horizontal label is supplied, which describes the reporting interval.
- Colors from palette number 1 are generated.

- The vertical label is supplied with the unit appropriate to the metric being graphed. For a CPU graph, the vertical label is percent (%), for a D_RATE it is rate/second, and BALANCE_SET is a number representing the number of processes.

17.3.1 Using a Graph Description File

Graph types and characteristics other than the default are specified using a graph description file. A graph description file is an ASCII file that describes graph type and characteristics, and is pointed to by the logical SPM$GRAPH_DESCRIPTION.

Table 17–1 describes the syntax of the graph description file.

Table 17–1 Graph Description File Syntax

Charactistic	Item	Description
GRAPH_ TYPE=		Specifies the type of graph.
	HISTOGRAM	Specifies a histogram graph.
	LINE	Specifies a line graph.
	PIE	Specifies a pie chart graph.
	STACKED_BAR	Specifies a stacked bar graph.
OPTIONS=		Specifies the type of option.
	[NO]BORDER= {CORNER | FULL}	Specifies the type of border.
	[NO]COLOR	Specifies whether the graph is in color shades or crosshatched.
	[NO]EXPANDED_PRINT	Specifies an enlarged graph for printing to sixel-capable devices (use only with /STYLE=SIXEL.)
	[NO]FILL	Specifies whether or not the area under a line graph is filled in.
	[NO]GRID={HORIZONTAL | VERTICAL | FULL}	Specifies the presence and attributes of a grid.
	[NO]HORIZONTAL_ LABEL="string"	Specifies text under the graph (limited to 40 characters.)
	[NO]LEGEND	Generates a graph legend that maps the colors or line types to the metrics being graphed.
	[NO]PALETTE = [1...12]	Describes the color palette used in the graphs (if printed or displayed on a device that does not support color, the graph is displayed or printed in shades of gray.)
	[NO]SHADOW	Highlights bar graphs with shadow behind the bars.

Table 17–1 (Cont.) Graph Description File Syntax

Charactistic	Item	Description
	[NO]STACK	Stacks line graphs in a manner similar to a stacked bar graph.
	[NO]SUBTITLE="string"	Subtitle text (limited to 40 characters).
	[NO]TITLE="string"	Title text (limited to 40 characters).
	[NO]VERTICAL_ LABEL="string"	Vertical label on the left side of the graph (limited to 20 characters).

Graph options are separated by commas and enclosed in parentheses as shown in the following example:

```
GRAPH_TYPE=STACKED_BAR
OPTIONS=(BORDER,SHADOW,LEGEND,TITLE="Disk Utilization")
```

To generate a shadowed bar graph with the title "Weekly CPU Usage for March 1989," perform the following steps:

1 Create a graph description file called BAR_GRAPH.TXT, containing the following:

```
GRAPH_TYPE=STACKED_BAR
OPTIONS=(SHADOW,
         TITLE="WEEKLY CPU USAGE FOR MARCH 1989")
```

2 Define the logical SPM$GRAPH_DESCRIPTION to point to the graph description file, as in this example:

```
$ ASSIGN BAR_GRAPH.TXT SPM$GRAPH_DESCRIPTION
```

3 Type the following command to generate the graph in ReGIS format from a history file:

```
$ PERFORMANCE REPORT=HISTORY/GRAPH/STYLE=REGIS/WEEKLY -
/SINCE=MAR-01-1988/BEFORE/APR-01-1988 HISTORY.DAT
```

17.4 Naming Conventions

Presentation graphs are generated one per file. The graph file names are based upon the conventions described in Table 17–2.

Table 17–2 Graph File-Naming Conventions

Graph Name Component	Value
disk:[directory]	The same as specified by the /OUTPUT qualifier, or the default directory.
"graph identifier"+	SPM$GRAPH is the name given by VAX SPM.

Table 17–2 (Cont.) Graph File-Naming Conventions

Graph Name Component	Value
"_graph type"+	The keyword name for the graphs. Several of the disk graph names have been truncated to conserve room for identifying the disk in the file name (examples are D_ALLOC, D_RESP, and D_UTIL).
"_disk/device name" +	$255$DUA1 (This item is present only in the name for disk or device graphs.)
"_graph number" +	Begins at 1 and is incremented for each graph of the same type.
".extension" +	.REGIS or .SIXEL, depending on the keyword selected for the /STYLE qualifier.
"_node or cluster name"	The name of the node or cluster.

For example, the first CPU, ReGIS style graph for node GREEN would be named as follows:

```
DISK1:USER_A:SPM$GRAPH_CPU_1.REGIS_GREEN
```

The first disk response time ReGIS style graph for the COLOR cluster would have the following name:

```
DISK1:USER_BSPM$GRAPH_D_RESP_$255$DUA1_1.REGIS_COLOR
```

17.5 Graph Selection Recommendations

The characteristics of the presentation graphs make them suited for graphing specific types of metrics. Graph types and recommendations are as follows:

Line

A line graph provides a breakdown of metrics over a large or small number of intervals.

Stacked Bar

The stacked bar graph provides a breakdown of items over a small interval. It is useful for graphing a limited number of columns as resource utilization over a year by months with 12 columns. Do not use this graph to represent a large number of intervals because this will obscure the breakdown of individual metrics.

The scale for the stacked bar graph is predetermined; therefore, it is not affected by a scale value specified for the /DISK, /DEVICES, and /GRAPH parameters.

Histogram

The histogram graph does not provide a breakdown of metrics and is useful when a breakdown is not necessary. For example, use a histogram graph to show overall CPU utilization.

Pie

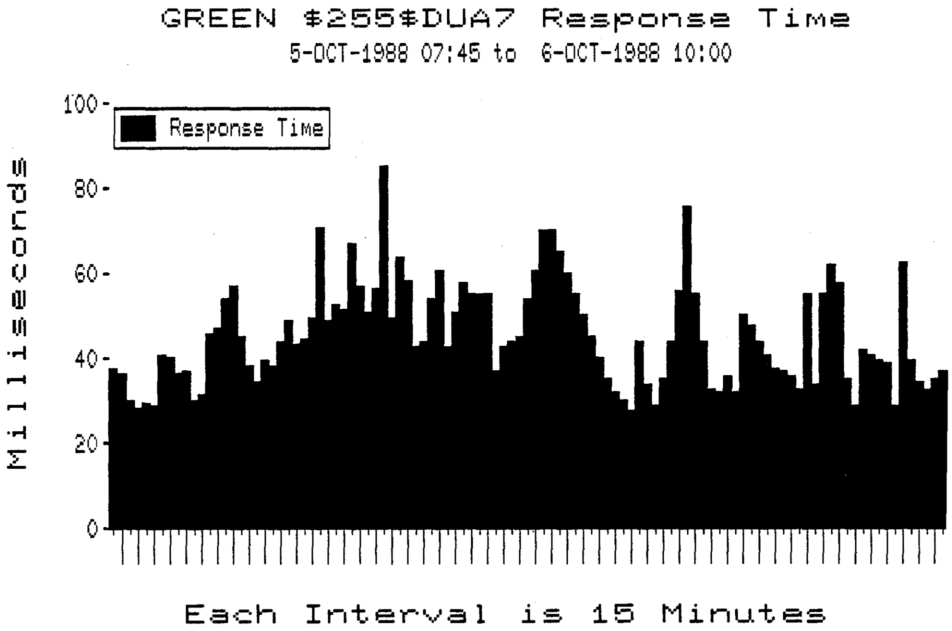
The pie graph provides a summary of multiple metrics. This graph is useful only for graphing resources with a variety of metrics such as CPU modes, or a system summary showing a number of resources.

Note that presentation graphs are not a replacement for ANSI style graphs, which are used primarily for system tuning.

17.6 Generating a Histogram Graph

Figure 17–1 is an example of a histogram graph showing disk respónse time.

Figure 17–1 Histogram Graph of Disk Response Time from Log File Data



Perform the following steps to generate a histogram graph similiar to the one in Figure 17–1:

1 Create a graph description file called HIST_GRAPH.TXT, which contains the commands in this example:

```
GRAPH_TYPE=HISTOGRAM
OPTIONS=NOBORDER
```

2 Define the logical SPM$GRAPH_DESCRIPTION to point to the file HIST_GRAPH.TXT as in this command:

```
$ DEFINE SPM$GRAPH_DESCRIPTION HIST_GRAPH.TXT
```

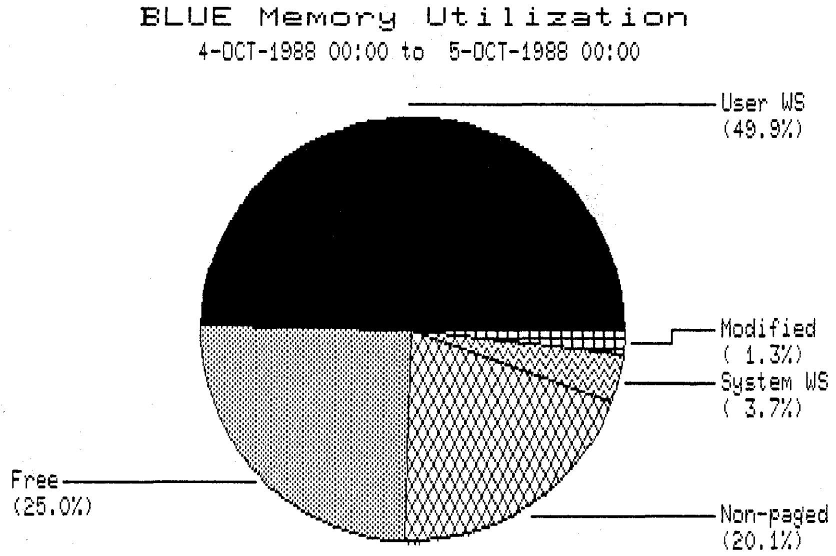
3 Type this command to generate the histogram graph:

```
$ PERFORMANCE REPORT=LOG_FILE LOG.DAT/STYLE=SIXEL/GRA=D_RESP/DISK=$255$DU
```

17.6.1 Generating a Pie Chart Graph

Figure 17–2 is an example of a pie chart graph showing memory utilization from all the data in a log file.

Figure 17–2 Pie Chart Graph of Memory Utilization from Log File Data



Perform the following steps to generate a pie chart graph similar to the one in Figure 17–2:

1 Create a graph description file called PIE_GRAPH.TXT, which contains the command in this example:

```
GRAPH_TYPE=PIE
```

2 Define the logical SPM$GRAPH_DESCRIPTION to point to the file PIE_GRAPH.TXT as in this command:

```
$ DEFINE SPM$GRAPH_DESCRIPTION PIE_GRAPH.TXT
```

3 Type this command to generate the pie chart graph:

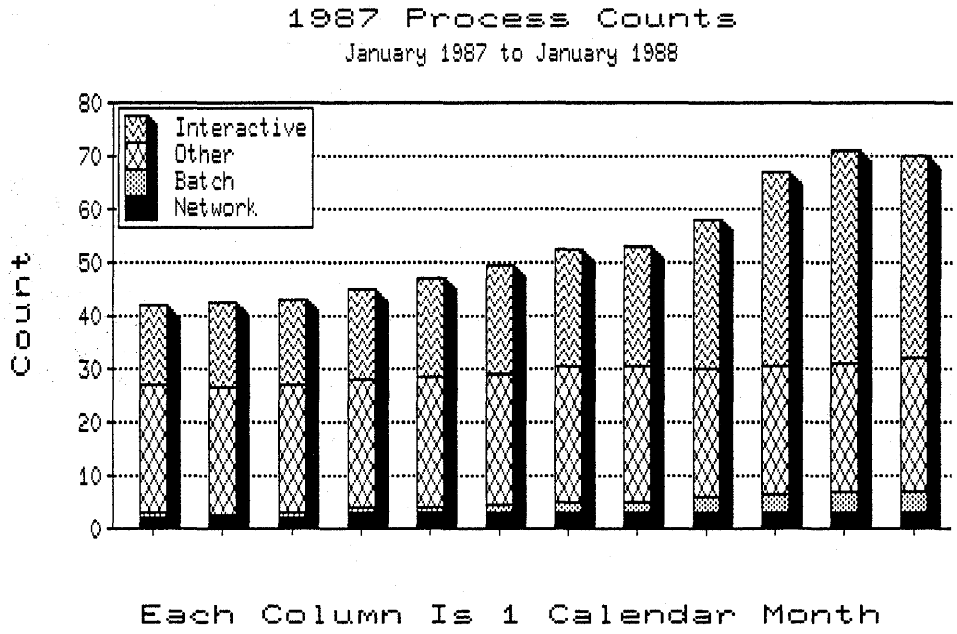
```
$ PERFORMANCE REPORT=LOG LOGFILE.DAT/STYLE=SIXEL/GRAPH=(MEMORY)/NOTABUL
```

17.6.2 Generating a Stacked Bar Graph

Figure 17–3 is an example of a stacked bar graph showing the number of processes over a year.

Perform the following steps to generate a stacked bar graph similar to the one in Figure 17–3 from data in a history file.

Figure 17–3 Stacked Bar Graph of Processes Over a Year



1 Create a dates file called DATES.DAT to define the months to be graphed. Section 16.6 describes how to create a dates file.

This is an example DATES.DAT file:

```
1-JAN-1987,1-FEB-1987
1-FEB-1987,1-MAR-1987
1-MAR-1987,1-APR-1987
1-APR-1987,1-MAY-1987
1-MAY-1987,1-JUN-1987
1-JUN-1987,1-JUL-1987
1-JUL-1987,1-AUG-1987
1-AUG-1987,1-SEP-1987
1-SEP-1987,1-OCT-1987
1-OCT-1987,1-NOV-1987
1-NOV-1987,1-DEC-1987
1-DEC-1987,1-JAN-1988
```

2 Define the logical SPM$DATES to point to the file DATES.DAT as in the following command:

```
$ DEFINE SPM$DATES DATES.DAT
```

3 Create a graph description file called BAR_GRAPH.TXT, which contains the commands in this example:

```
GRAPH_TYPE=STACKED_BAR
OPTIONS=(GRID=HORIZONTAL,
  SHADOW,
  TITLE="1987 Process Counts",
  SUBTITLE="January 1987 to January 1988",
         HORIZONTAL_LABEL="Each Column is 1 Calendar Month")
```

4 Define the logical SPM$GRAPH_DESCRIPTION to point to the file BAR_GRAPH.TXT as in the following command:

```
$ DEFINE SPM$GRAPH_DESCRIPTION BAR_GRAPH.TXT
```

5 Type this command to generate the bar graph:

```
$ PERFORMANCE REPORT=HISTORY HIST.DAT/STYLE=SIXEL/GRAPH=(PROCESS) -
/NOTABULAR/GENERAL/SHIFT=PRIME
```

18 Reporting Trend Analysis Data

This chapter applies the commands described in Chapter 16 to generate a variety of trend analysis reports and graphs.

One point to remember when reporting trend analysis data is that configuration changes to your system can produce changes in resource utilization on a graph. For example, if CPU utilization is usually 80%, CPU utilization may decline sharply after a larger processor is installed. The decline in CPU utilization reflects the increase in available CPU.

This chapter describes:

- Performing prerequisites to trend analysis reporting
- Defining the scope of reports
- Reporting on each month
- Reporting on each fiscal month
- Graphing a year by weeks
- Graphing one year's typical week
- Graphing a year by months
- Graphing data for the most active disks over a year
- Graphing other metrics
- Producing a stacked bar graph of monthly CPU utilization

18.1 Prerequisites

Before you can begin to produce trend analysis reports, you need:

- A history file containing at least a year's[1] worth of data
- Information about your company:
  - — Normal working hours
  - — Normal working days
  - — List of company holidays
  - — Normal length of a reporting month

[1] Although the examples in this chapter report on a history file containing more than a year's worth of data, you can create variations of these reports and graphs using less data to evaluate resource usage trends.

18.2 Defining the Scope of Reports

The first step in reporting trend analysis data is to define the scope of the report to ensure that meaningful data is reported. Defining the scope of the report includes these tasks:

- Defining hours in the day as prime using the /P_HOURS qualifier
- Defining days of the week as having prime hours using the /P_DAYS qualifier
- Defining days your company designates as holidays as non-prime using a holidays file
- Defining custom months to accommodate your company's reporting structure

The sections below show one way to define the scope of a report. Read the *VAX SPM Reference Manual* or Section 16.3 of this guide for more information about the /P_HOURS, /P_DAYS qualifiers, and the holidays and dates files.

18.2.1 Defining Hours in the Day as Prime Hours

Select prime hours for reporting by including the /SHIFT=PRIME qualifier in the command line. Define prime hours using the /P_HOURS qualifier.

The hours you define as prime depend on how your system is used. If your system has mostly interactive users, prime time is the hours they work. For example, if your work day is from 8 a.m. to 11 a.m. and 1 p.m. to 6 p.m., define prime hours as follows:

```
/P_HOURS=(8-10,13-17)
```

On the other hand, if your system is used around the clock to run batch jobs, you may specify prime hours for 24 hours of the day as in the following example:

```
/P_HOURS=(0-23)
```

The default for prime hours is the range 8–16, or 8 a.m. to 5 p.m. If these hours decribe your work day accurately, you do not have to specify any value for the P_HOURS qualifier.

To limit the reporting to prime hours, however, you must include the /SHIFT=PRIME qualifier in the command line.

18.2.2 Defining Days of the Week as Prime and Nonprime

Define weekdays as either prime or nonprime by using the /P_DAYS qualifier.

The days of the week you select as prime depend on your company's workweek. For example, if your workweek begins on Tuesday and continues through Saturday, you would define this workweek using the /P_DAYS qualifier as follows:

```
/P_DAYS=(NOMONDAY,SATURDAY)
```

A day is designated as nonprime by preceding it with NO.

By default, MONDAY through FRIDAY have prime hours, while SATURDAY and SUNDAY are nonprime time. If this describes your workweek, do not specify values for the /P_DAYS qualifier or include it in your command line. As is the case with prime hours, however, you need to include the /SHIFT=PRIME qualifier on the command line.

18.2.3 Defining Holidays as Nonprime

Designate the holidays observed by your company as nonprime days by performing the following steps:

1 Create a file called HOLIDAYS.TXT containing the dates of your holidays.

   Dates are specified one to a line in the following format:

   ```
   dd-mmm-yyyy
   ```

   Ranges of dates are separated by a comma. You can specify as many as 60 dates in the holidays file.

   This is an example of dates specified in a holidays file:

   ```
   01-JAN-1988
   30-MAY-1988
   04-JUL-1988
   05-SEP-1988
   26-NOV-1988
   25-NOV-1988
   25-DEC-1988
   27-DEC-1988,01-JAN-1989
   02-JAN-1989
   29-MAY-1989
   03-JUL-1989
   04-JUL-1989
   04-SEP-1989
   23-NOV-1989
   24-NOV-1989
   25-DEC-1988,30-DEC-1988
   ```

   Note that in the above example, ranges of dates represent a week's vacation observed by company over Christmas.

2 Define the logical SPM$HOLIDAYS to point to the file you created. For a holidays file HOLIDAYS.TXT, use the following command:

   ```
   $ DEFINE SPM$HOLIDAYS HOLIDAYS.TXT
   ```

3 Include the /SHIFT=PRIME qualifier in your command line.

18.2.4 Defining Months to Accommodate Your Reporting Structure

The /MONTHLY qualifier specifies a month as the reporting unit and defines a month as follows:

```
Begins at 00:00 on the first day of the month
Ends at 00:00 hours on the first day of the following month
```

```
For example:
The month of September begins on
01-SEP-1988:00:00 and ends 01-OCT-1988:00:00
```

Depending on the reporting structure of your company, you may wish to define fiscal months or months with an equal number of days to replace the default month. Note, however, that when using the /MONTHLY qualifier with an SPM$DATES file, the date range in the dates file cannot exceed 35 days for a month.

Perform the following steps to define custom months for reporting:

1. Create a file called DATES.TXT to define the beginning and end of each month. Each line in the dates file has the following format:

   ```
   dd-mmm-yyyy:hh:ss:cc,dd-mmm-yyyy:hh:ss:cc
   ```

   The following example defines beginning and ending dates for months in a fiscal year:

   ```
   28-JUN-1988,25-JUL-1988
   25-JUL-1988,22-AUG-1988
   22-AUG-1988,26-SEP-1988
   26-SEP-1988,24-OCT-1988
   24-OCT-1988,21-NOV-1988
   21-NOV-1988,26-DEC-1988
   26-DEC-1988,23-JAN-1989
   23-JAN-1989,20-FEB-1989
   20-FEB-1989,26-MAR-1989
   27-MAR-1989,23-APR-1989
   23-APR-1989,21-MAY-1989
   21-MAY-1989,24-JUN-1989
   ```

2. Define the logical SPM$DATES to point to the dates file as in the following example:

   ```
   $ DEFINE SPM$DATES DATES.TXT
   ```

3. Include the /MONTHLY qualifier in the command line.

18.3 Reporting on Each Month

This section provides an example of how to produce the following reports and graphs:

- One final statistics tabular report for each month in the year
- One process graph of the average number of processes for each month in the year

The reports are produced using this command:

```
$DEFINE SPM$HOLIDAYS HOLIDAYS.TXT
$ PERFORMANCE REPORT=HISTORY HISTORY.DAT -
 /GRAPH=(PROCESS) -
 /P_HOURS=(8-10,13-17) -
 /P_DAYS=(NOMONDAY,SATURDAY) -
 /SHIFT=PRIME -
 /MONTHLY -
 /BEFORE="01-JAN-1989"/SINCE="01-JAN-1988"
```

18.4 Reporting on Each Fiscal Month

This section provides an example of how to produce the following reports and graphs:

- One final statistics tabular report for each month in the year as defined in the dates file
- One process graph of the average number of processes for each month in the year as defined by the dates file, with each column in the graph representing statistics for 8 hours

The reports are produced using this command:

```
$ DEFINE SPM$DATES DATES.TXT
$ DEFINE SPM$HOLIDAYS HOLIDAYS.TXT
$ PERFORMANCE REPORT=HISTORY HISTORY.DAT -
 /GRAPH=(PROCESS) -
 /P_HOURS=(8-10,13-17) -
 /P_DAYS=(NOMONDAY,SATURDAY) -
 /SHIFT=PRIME -
 /MONTHLY
```

In the above command, it is not necessary to specify a reporting period because one is already defined by the range of dates in the dates file. If, however, the dates file covered a longer period than you wanted to report, you would need to limit the reporting period using the /BEFORE and /SINCE qualifiers.

18.5 Graphing a Year by Weeks

This section provides an example of how to produce the following reports and graphs:

- One final statistics tabular report for the entire year
- A single graph of the average number of processes for an entire year with each column in the graph representing one week

The reports are produced using this command:

```
$ DEFINE SPM$HOLIDAYS HOLIDAYS.TXT
$ PERFORMANCE REPORT=HISTORY HISTORY.DAT -
 /GRAPH=(PROCESS) -
 /P_HOURS=(8-10,13-17) -
 /P_DAYS=(NOMONDAY,SATURDAY) -
 /SHIFT=PRIME -
 /GENERAL=(REPORT_INTERVAL=(10080),INTERVAL=52) -
 /BEFORE="01-JAN-1989"/SINCE="01-JAN-1988"
```

In the above command, the /GENERAL qualifier is used as follows:

- Specify a week as the reporting unit by specifying a value of 10080 for the REPORT_INTERVAL keyword. The number of minutes in a week, (10080) is obtained by multiplying 60×24×7.
- Specify a reporting interval of 52 by using the /INTERVAL qualifier.

18.6 Graphing One Year's Typical Week

This section provides an example of how to produce the following reports and graphs:

- One final statistics tabular report of the typical week for a year
- One graph showing the average number of processes for a typical week in the year with each column in the graph representing statistics for one day of the typical week

The reports are produced using this command:

```
$ DEFINE SPM$HOLIDAYS HOLIDAYS.TXT
$ PERFORMANCE REPORT=HISTORY HISTORY.LOG -
 /GRAPH=(PROCESS) -
 /SHIFT=PRIME -
 /P_HOURS=(8-10,13-17) -
 /P_DAYS=(NOMONDAY,SATURDAY) -
 /WEEKLY -
 /AVERAGE -
 /BEFORE="01-JAN-1989"/SINCE="01-JAN-1988"
```

In the above command:

- The /WEEKLY qualifier specifies a week as the reporting unit.
- The /AVERAGE qualifier specifies that data for all reporting units (weeks) is averaged together over the reporting year to produce an average or typical week.

18.7 Graphing a Year by Months

This section provides an example of how to produce the following reports and graphs:

- One final statistics tabular report for the entire year
- A single graph of the average number of processes for an entire year with each column in the graph representing one month

The reports are produced using this command:

```
$ DEFINE SPM$DATES DATES.TXT
$ DEFINE SPM$HOLIDAYS HOLIDAYS.TXT
$ PERFORMANCE REPORT=HISTORY HISTORY.DAT -
 /GRAPH=(PROCESS) -
 /P_HOURS=(8-10,13-17) -
 /P_DAYS=(NOMONDAY,SATURDAY) -
 /SHIFT=PRIME -
 /GENERAL=(INTERVAL=12)
```

In the above command:

- DEFINE SPM$DATES DATES.TXT ensures that months defined in the dates file are used to describe months.
- /GENERAL=(INTERVAL=12) specifies a reporting interval of 12.

18.8 Graphing Data for the Most Active Disks Over a Year

This section provides an example of how to produce the following reports and graphs:

- One final statistics tabular report for each month in the year
- One disk rate graph per month for only those disks that have more than 15 I/Os per second over 30% of the time with each column in the graph representing one month

The reports are produced using this command:

```
$ DEFINE SPM$HOLIDAYS HOLIDAYS.TXT
$ PERFORMANCE REPORT=HISTORY HISTORY.DAT -
 /P_HOURS=(8-10,13-17) -
 /P_DAYS=(NOMONDAY,SATURDAY) -
 /SHIFT=PRIME -
 /GRAPH=(D_RATE:15) -
 /SELECT=GREATER_THAN:30 -
 /MONTHLY -
 /BEFORE="01-JAN-1989"/SINCE="01-JAN-1988"
```

In the above command:

- /GRAPH=(D_RATE:15) specifies a threshold of 15 for disk statistics.
- /SELECT=GREATER_THAN:30 specifies that graphs will be generated only for disks with more than 15 I/Os per second more than 30% of the time.

18.9 Graphing Other Metrics

This section provides an example of how to produce the following reports and graphs:

- A single graph of the average number of page faults for an entire year with each column in the graph representing one week

The reports are produced using this command:

```
$ DEFINE SPM$HOLIDAYS HOLIDAYS.TXT
$ PERFORMANCE REPORT=HISTORY HISTORY.DAT -
 /NOTABULAR -
 /GRAPH=(PAGE_FAULTS) -
 /SHIFT=PRIME -
 /P_HOURS=(8-10,13-17) -
 /P_DAYS=(NOMONDAY,SATURDAY) -
 /GENERAL=(REPORT_INTERVAL=(10080),INTERVAL=52) -
 /BEFORE="01-JAN-1989"/SINCE="01-JAN-1988"
```

In the above command:

- /NOTABULAR supresses the generation of tabular reports.

- /GRAPH=(PAGE_FAULTS) specifies that page fault statistics will be graphed.
- /GENERAL=(REPORT_INTERVAL=10080,INTERVAL=52) specifies the reporting interval as 52 and the reporting unit as one week (in minutes).

18.10 Producing a Stacked Bar Graph of Monthly CPU Utilization

This section provides an example of how to produce the following presentation graph: a stacked bar graph type showing the CPU utilization for an entire year with each bar in the graph representing one month.

Follow the instructions below:

1 Create a graph description file called BAR_GRAPH.TXT, which contains the commands in the following example:

```
GRAPH_TYPE=STACKED_BAR
OPTIONS=(GRID=HORIZONTAL,
  SHADOW,
  TITLE="CPU Utilization"
  SUBTITLE="January 1988 to January 1989",
         HORIZONTAL_LABEL="Each Column is 1 Fiscal Month")
```

2 Define the logical SPM$GRAPH_DESCRIPTION to point to the file BAR_GRAPH.TXT as in the following command:

```
$ DEFINE SPM$GRAPH_DESCRIPTION BAR_GRAPH.TXT
```

The graph is produced using this command:

```
$ DEFINE SPM$GRAPH_DESCRIPTION BAR_GRAPH.TXT
$ DEFINE SPM$DATES DATES.TXT
$ DEFINE SPM$HOLIDAYS HOLIDAYS.TXT
$ PERFORMANCE REPORT=HISTORY HISTORY.DAT -
 /NOTABULAR -
 /GRAPH=(CPU)/STYLE=REGIS -
 /P_HOURS=(8-10,13-17) -
 /P_DAYS=(NOMONDAY,SATURDAY) -
 /SHIFT=PRIME -
 /GENERAL=(INTERVAL=12)
```

In the above command:

- /NOTABULAR suppresses the generation of tabular reports.
- /GRAPH=(CPU) specifies a graph of CPU statistics.
- /STYLE=REGIS specifies a presentation quality graph in ReGIS format.

19 The Event Trace Facility

This chapter describes the Event Trace Facility (ETF). The ETF is used to:

- Monitor locations in system virtual address space.
- Collect data at these locations using routines you have written.

19.1 Introduction

The SPM Event Trace facility is a collection of system services that can be used to monitor and record events occurring in the VMS executive. It can be called from a number of high level languages such as BLISS, PASCAL, C and BASIC, as well as VAX MACRO. The languages that call ETF are determined when VAX SPM is installed.

The facility is general purpose in that the user defines the locations in system virtual address space (trace points) to monitor. Moreover, the user writes a data collection routine for each trace point, thus collecting whatever data is desired at the trace point.

The facility provides the routines needed to initialize itself, define and activate trace points, pass control to each user-defined data collection routine, and perform buffer and record management.

Warning:

- **The VMS algorithms and data structures are subject to change as the executive is updated.**
- **There is no guarantee that user-written data collection code will continue to work with new releases of VMS.**
- **Improperly coded data collection routines may crash the running VMS system.**

VAX SPM reports are useful data for general system tuning and trend analysis applications. There are some rather complex forms of data that VAX SPM does not collect or report, however, and for which the VMS operating system does not provide access. For example, information about the difference between the sizes of nonpaged pool requested and the size actually allocated is not accessible using system tuning or trend analysis reports provided by VAX SPM.

For collecting complex data not ordinarily provided by VAX SPM or the VMS operating system itself, a different sort of tool is needed. This tool is the Event Trace Facility (ETF).

While investigating a CPU bottleneck, if the PC analysis reveals that a lot of CPU time is being spent in a private device driver, the ETF could be used to monitor the driver to learn which sections of code are using the CPU time.

19.2 Using the Event Trace Facility

The following privileges and quotas are required to use the Event Trace facility:

- Process privileges—CMKRNL
- Nonpaged pool—3300 bytes + size of I/O buffers + size of user-written executive trace routines
- Two (2) global sections and global pages for the ETF privileged sharable image, SYS$SHARE:SPM$ETFSHR.EXE

To use the Event Trace facility to collect data, the following steps are required:

1. Identify the events (system virtual address) where event data is to be collected.
2. Write the executive trace routines to collect the data.
3. Initialize the trace facilities to define the buffering requirements and declare the executive trace routines.
4. Define each trace point by a trace point ID and associate an address within the executive with the trace point.
5. Activate the trace points.
6. Define the amount of data required and what to do when sufficient data is collected.
7. Shut down the facility.

Figure 19–1 is an example of a FORTRAN program that calls the ETF. There is an example of a program that uses the ETF in SPM$EXAMPLES:SPMETF_EXAMPLE.COM.

Figure 19–1 User-Written Routine That Calls ETF

```
 PROGRAM EVENT_TRACE_EXAMPLE
         .              .                              .
         .              .                              .
         .              .                              .
 status = LIB$GET_EF( SIGNAL_EFN )
         .              .                              .
         .              .                              .
         .              .                              .
 ROUTINE_SIZE = %LOC( POOL_MONITOR_END ) - %LOC( POOL_MONITOR )
      status = spm$cretf( %LOC(POOL_MONITOR),
     1      %VAL(ROUTINE_SIZE),
     1      %VAL(2),
     1      %VAL(512),
     1      %VAL(22),
     1      %VAL(SIGNAL_EFN))
         .              .                              .
         .              .                              .
         .              .                              .
 status = spm$deftp( %VAL(1),
     1       %LOC(EXE$ALONONPAGED),
     1       %VAL(0),
     1       %LOC(POOL_MONITOR),
     1       %REF(SVARET))

 status = spm$acttp( )
         .              .                              .
         .              .                              .
         .              .                              .
 END

---- POOLMON_SUB.MAR --

; COPYRIGHT (c) 1988 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
; ALL RIGHTS RESERVED.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

;++
; Facility:
;   VAX SPM EXAMPLES
;
```

Figure 19–1 Cont'd. on next page

Figure 19–1 (Cont.) User-Written Routine That Calls ETF

```
 .TITLE POOLMON_SUB - Pool Monitor
       .IDENT /X-1/
 .LIBRARY /SYS$LIBRARY:SPM$EVENT_TRACE.MLB/
 .LIBRARY /SYS$LIBRARY:LIB.MLB/
 .SBTTL DEFINITIONS
;+
;
;  User defined data collection routine
;  This code gets copied into pool allocated by ETF during
;  the $CRETF call.
;
; This routine performs the following actions:
;
; Calls ETF at the provided entrypoint to get a free ETF RECORD
; in which to write data.  Note if there is a full buffer this
; call results in the setting of the EVENT FLAG specified on
; the $CRETF call.
;
; With the allocated record, this routine saves:
; - The trace point id
; - The ETF provided time stamp
; - The PID of the current process at the time of the Trace Point
; - The Physical CPUid of the CPU taking the trace point
; - The size in bytes of pool being requested.
;
;-
POOL_MONITOR::                           ; START OF DATA COLLECTION CODE
 PUSHR #^M<R0,R2,R4>
 PUSHL   #PMON$K_RECSIZE          ; Record length desired
         .             .                           .
         .             .                           .
         .             .                           .
 MOVL    CPU$L_PHY_CPUID(R0),PMON$L_CPU(R2); CPUID OF CPU
 MOVL    R1,PMON$L_REQ(R2)            ; Save the Size.
10$: POPR    #^M<R0,R2,R4>                ; Restore Regs
 RSB                                  ; Return
POOL_MONITOR_END::
 .END
```

The following commands would be used to build and link this example:

```
$ FORTRAN EXAMPLE
$ MACRO POOLMON_SUB
$ LINK EXAMPLE,POOLMON_SUB,SYS$SYSTEM:SYS.STB/SEL,SYS$INPUT/OPT-
SYS$SHARE:SPM$ETFSHR/SHARE
```

Figure 19–2 illustrates how a user-written event trace program (UWTP) calls the ETF service to use the Event Trace Facility (ETF).

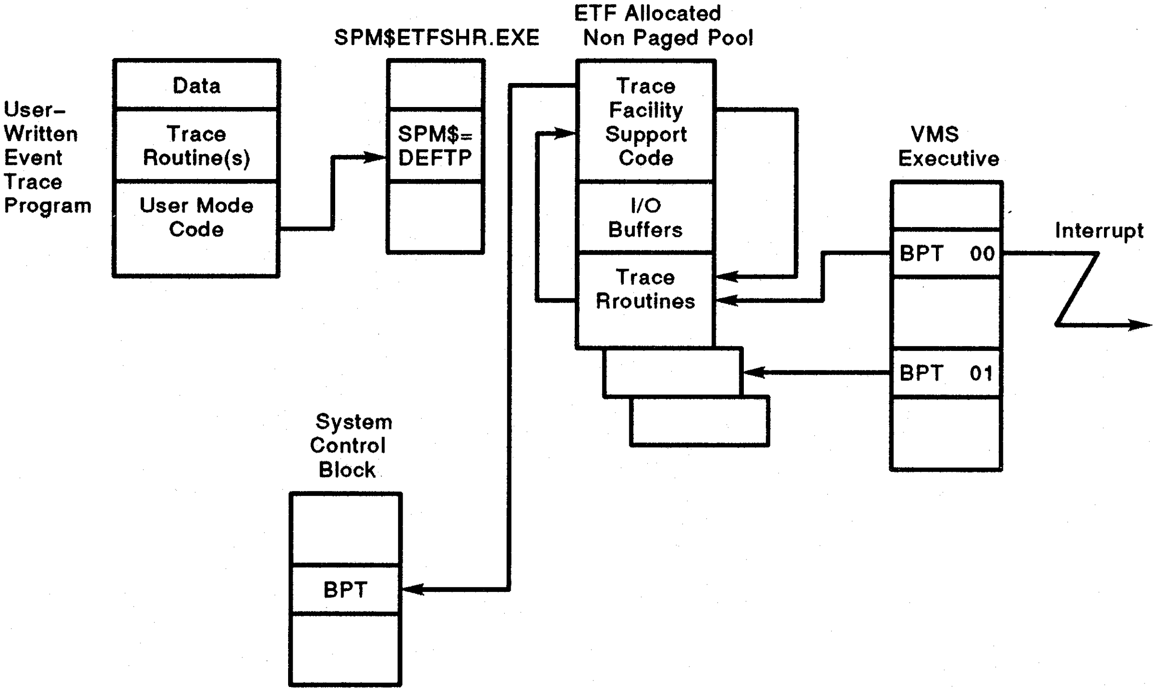
The ETF is comprised of service routines. These service routines create the facility, define and activate trace points, receive and return data buffers, and upon completion, delete the facility.

When the UWTP calls SPM$DEFTP, entries are made in the nonpaged pool allocated to the ETF. The UWTP activates the trace points by a call to SPM$ACTTP. The ETF replaces the BPT vector in the System Control Block (SCB) and places breakpoints in the VMS executive at the trace point locations given by the SPM$DEFTP service calls.

When the hardware encounters a trace point (and executes the BPT instruction), the ETF support code is executed by an interrupt through the SCB.

The ETF support code calls the user-written trace routines, which reside in nonpaged pool. Next, the ETF support code executes the "original" instruction, and returns from the interrupt.

Figure 19–2 Example Showing Trace Facility



A description of each of the SPM event trace services are found in the *VAX SPM Reference Manual*.

20 Charging for System Usage

This chapter provides the following information about the use of the VAX SPM Charge facility:

- Introduction to the Charge facility
- Instructions for specifying resource prices
- Instructions for invoking the Charge facility
- Instructions for reporting disk usage charges
- Instructions for varying the content of Charge reports

20.1 Introduction to the Charge Facility

The VAX SPM Charge facility uses VMS accounting data to produce system utilization reports. These reports show monetary amounts charged for various types of system resources based upon unit prices provided by the user. You can use the Charge report as an itemized bill or as a general resource utilization report.

With the VAX SPM Charge facility you can:

- Produce a single report from multiple accounting files created by the VMS ACCOUNTING utility on a single node or a VAXcluster system.
- Charge different prices for resources in different accounting files.
- Charge different prices for printed pages according to printer queue names and accounting files.
- Charge different prices for disk usage according to volume names and the disk usage information provided by the VMS ANALYZE/DISK_STRUCTURE/USAGE utility.
- Generate reports on the basis of single jobs, job types, UICs, users accounts, and grand totals.

The Charge facility uses information from accounting log files named SYS$MANAGER:ACCOUNTNG.DAT. The Charge facility can read data directly from open accounting files on different nodes.

Prior to reporting on accounting data, you must specify the unit prices charged for each system resource. These amounts can be entered interactively from the terminal, but it is usually more convenient to place them in a price file using a text editor.

To report disk utilization, use the VMS ANALYZE/DISK_STRUCTURE/USAGE DCL command to generate a disk usage file for each disk on which you wish to report. Append the disk usage files into one file for processing by the VAX SPM Charge facililty.

The PERFORMANCE REPORT=CHARGE command processes the accounting and disk usage data with the price file containing resource prices. It outputs a report showing a detailed breakdown of charges by individual job or process, job type, UIC, user name, account name, and grand totals.

To summarize, using the VAX SPM Charge facility involves these steps:

1 Determine the amounts to be charged for each type of system resource and place these in a price file as described in Section 20.2.

2 If you wish to report on disk allocation charges, gather disk usage information as described in Section 20.4.

3 Use the PERFORMANCE REPORT=CHARGE command described in Section 20.3 to process the accounting and disk data with amounts in the price file, and generate a report.

20.2 Specifying Resource Prices

Prices for resources may be specified in two ways: in a price file and from the terminal.

The rules for constructing a price file are summarized as follows:

- Each line can begin in any column in the price file line.
- The list is comprised of keywords and their values. Most keywords may be preceded by identification information.
- No more than one keyword specification can be placed on the same line, and no single keyword specification may be split across lines.
- As in DCL command files, comment text is preceded by an exclamation point (!).

To enter prices from the terminal, type the charge command and omit the charge-file parameter from the command line. If the disk-usage parameter is specified, replace the charge-file parameter with two double quotes (" ").

The following message is displayed:

```
Enter price and control characters below:
```

Enter keywords and values according to the following format:

```
KEYWORD=VALUE
```

Example:

```
TITLE="Charges for Today"<CR>
PAGEFAULT_PRICE=80<CR>
```

20.2.1 Overview and Example Price File

Price file keywords may be divided into two categories: title and resource, as shown in Table 20–1. Use the TITLE keyword to specify a title for your report and the resource keywords to specify prices for system resources.

Table 20–1 Price List Keywords and Default Values

Keyword	Meaning	Default Value
Title Keyword		
TITLE	Optional Report Title	Title line is not printed
Resource Keywords		
BUFFEREDIO_PRICE	Buffered I/O Operations	0.00000
CPUSEC_PRICE	CPU Time	0.00000
DIRECTIO_PRICE	Direct I/O Operations	0.00000
ELAPSEDSEC_PRICE	Elapsed Time	0.00000
FAULTIO_PRICE	Page Read I/O Operations	0.00000
GET_PRICE	Symbiont GET Operations	0.00000
IMAGE_PRICE	Image Activations	0.00000
MOUNTVOL_PRICE	Volume Mounts	0.00000
PAGE_PRICE	Pages Printed by Symbiont	0.00000
PAGEFAULT_PRICE	Page Faults	0.00000
PROCESS_PRICE	Process Creations	0.00000
QIO_PRICE	Symbiont QIO Operations	0.00000
PRINTJOB_PRICE	Print Jobs	0.00000
DISKBLOCK_PRICE	Allocated Block per UIC	0.00000

Figure 20–1 shows an example price file that specifies all allowed keywords. The values shown are suggested as a starting point for the creation of a price list for your system.

20.2.2 Specifying a Value for the Report Title Keyword

The report title keyword is TITLE. TITLE is an 80-character string that is printed as the title of each page of the Charge report.

If a string is not specified, this line is not printed. If more than 80 characters are specified, the title is truncated to 80 characters. Enclose the string in single quotes (') as shown in the following example:

```
TITLE = 'January's Color Cluster Resource Charges'
```

20.2.3 Specifying Values for Resource Keywords

The Charge utility can produce a report from a number of accounting files with different resource prices for each accounting file. This is accomplished by specifying monetary values and identfication information for resource keywords.

Table 20–2 describes the resource keywords.

A monetary value assigns a price to a particular resource and is specified according to the following format:

```
resource_price = value
```

The following is an example:

```
 BUFFEREDIO_PRICE = 0.00010
```

If no value is specified, the default value of zero is assumed for that resource in all accounting files.

Identification information distinguishes the accounting file, printer, or disk for which a resource price is to apply. A prefix is a type of identification information which is applied to all resource keywords except DISKBLOCK_PRICE. The purpose of a prefix is to link resource prices in the price file to specific accounting files. Prefixes are specified with resource keywords in the price file and with accounting files in the CHARGE command line. Section 20.3 describes the use of prefixes in the CHARGE command.

Table 20–2 Resource Keywords

Keyword	Description
BUFFEREDIO_PRICE	The price for each buffered I/O operation.
CPUSEC_PRICE	The price for each second of CPU time used.
DIRECTIO_PRICE	The price for each direct I/O operation.
ELAPSEDSEC_PRICE	The price for each second of elapsed time between process creation and process deletion.
FAULTIO_PRICE	The price for each disk read operation to bring into memory a page fault cluster to satisfy a page fault. Typically, this has the same value as DIRECTIO_PRICE.
GET_PRICE	The price of each RMS $GET done by a print symbiont (on the user's behalf) to read a file being printed.
IMAGE_PRICE	The price for each image activation.
MOUNTVOL_PRICE	The price for each volume mount operation (handling cost).
PAGEFAULT_PRICE	The price for each page fault.
PROCESS_PRICE	The price for each process creation.
QIO_PRICE	The price for each $QIO done by a print symbiont (on the user's behalf) to send characters to the output device. Typically, this has the same value as BUFFEREDIO_PRICE.
PRINTJOB_PRICE	The price for each print job on the specified printer queue.

Table 20–2 (Cont.) Resource Keywords

Keyword	Description
PAGE_PRICE[1]	The price for each page of printed output for the specified printer.
DISKBLOCK_PRICE[2]	The price for each allocated block per UIC for the specified disk.

[1] The PAGE_PRICE keyword may be preceded by a printer queue name as well as a prefix.

[2] The DISKBLOCK_PRICE keyword may be preceded by a volume name only and may not be preceded by a prefix.

Identification information is specified depending on how resources are charged. Resources may be charged in the following ways:

- Identical resource prices may be charged for all accounting files.
- Different resource prices may be charged according to accounting files based on user-defined criteria such as processor type.
- Different prices may be charged for print pages according to printer queue name and accounting file.
- Different prices may be charged for disks based on volume names.

Each of these approaches is described in the sections below.

20.2.3.1 Specifying Identical Prices for Accounting Files

Figure 20–1 is an example of a price file with one set of prices that may be applied to a number of accounting files. There are no prefixes in this file; therefore, prefixes are not necessary in the CHARGE command line. If a prefix is specified for an accounting file in the CHARGE command, a warning message is displayed and charge file prices are applied to that accounting file.

Figure 20–1 Price File Showing One Price for All Accounting Files

```
TITLE  = 'September's System Usage Charges'
BUFFEREDIO_PRICE =  0.00010
CPUSEC_PRICE     =  0.01000
DIRECTIO_PRICE   =  0.00050
ELAPSEDSEC_PRICE =  0.00005
FAULTIO_PRICE    =  0.00050
GET_PRICE        =  0.00010
IMAGE_PRICE      =  0.01000
MOUNTVOL_PRICE   =  1.00000
PAGE_PRICE       =  0.02500
PAGEFAULT_PRICE  =  0.00001
PROCESS_PRICE    =  0.10000
QIO_PRICE        =  0.00010
DISKBLOCK_PRICE  =  0.05000
PRINTJOB_PRICE   =  1.00000
```

20.2.3.2 Specifying Different Prices for Accounting Files

A pricing scheme in which a resource's price can vary according to the accounting file is accomplished through the use of prefixes. As stated above, prefixes are specified with resource keywords in the price file and with accounting files in the CHARGE command. Section 20.3 describes specifying prefixes in the CHARGE command.

In the price file, different resource prices may be specified for each prefix according to the following format:

```
prefix::resource_keyword = value
```

The following is an example:

```
8800::BUFFEREDIO_PRICE = 0.20
2000::BUFFEREDIO_PRICE = 0.10
GREEN::BUFFEREDIO_PRICE = 0.30
```

For an example of the use of prefixes, consider the Color VAXcluster system described in Table 20–3.

Table 20–3 Color VAXcluster System Nodes and Processor Types

Node Name	VAX Processor Type
PINK	VAX 8800
AQUA	VAX 2000
PURPLE	VAX 8800

Resources on nodes PINK and PURPLE may be charged at a higher rate than the resources on node AQUA. Therefore, each resource in the price file would be specified with one price for the prefix 8800 and one price for the prefix 2000 as shown in Figure 20–2.

When /PREFIX=8800 is specified with accounting files in the CHARGE DCL command line, amounts for all resources preceded by the prefix 8800 in the price file are used to calculate values for those accounting files. Likewise, when /PREFIX=2000 is specified with accounting files, amounts for all resources preceded by the prefix 2000 are used to calculate values for those accounting files.

Prices for resources without prefixes are applied to an accounting file for either of the following conditions:

- There is no prefix on the command line.
- The prefix specified does not correspond to any prefixes in the price file.

Figure 20–2 Price File Showing Amounts for Two Prefixes

```
TITLE  = 'System Resource Usage Charges for COLOR Cluster'
! Default Prices
BUFFEREDIO_PRICE =  0.00010
CPUSEC_PRICE     =  0.01000
DIRECTIO_PRICE   =  0.00050
ELAPSEDSEC_PRICE =  0.00005
FAULTIO_PRICE    =  0.00050
GET_PRICE        =  0.00010
IMAGE_PRICE      =  0.01000
MOUNTVOL_PRICE   =  1.00000
PAGE_PRICE       =  0.02500
PAGEFAULT_PRICE  =  0.00001
PROCESS_PRICE    =  0.10000
QIO_PRICE        =  0.00010
DISKBLOCK_PRICE  =  0.05000
PRINTJOB_PRICE   =  1.00000
!
!Prices for nodes PINK and PURPLE
!
8800::BUFFEREDIO_PRICE =  0.00020
8800::CPUSEC_PRICE     =  0.02000
8800::DIRECTIO_PRICE   =  0.00060
8800::ELAPSEDSEC_PRICE =  0.00006
8800::FAULTIO_PRICE    =  0.00060
8800::GET_PRICE        =  0.00020
8800::IMAGE_PRICE      =  0.02000
8800::MOUNTVOL_PRICE   =  2.00000
8800::PAGE_PRICE       =  0.05000
8800::PAGEFAULT_PRICE  =  0.00002
8800::PROCESS_PRICE    =  0.20000
8800::QIO_PRICE        =  0.00020
      DISKBLOCK_PRICE  =  0.06000
8800::PRINTJOB_PRICE   =  2.00000
!
!Prices for node AQUA
!
2000::BUFFEREDIO_PRICE =  0.00009
2000::CPUSEC_PRICE     =  0.00900
2000::DIRECTIO_PRICE   =  0.00040
2000::ELAPSEDSEC_PRICE =  0.00004
2000::FAULTIO_PRICE    =  0.00040
2000::GET_PRICE        =  0.00009
2000::IMAGE_PRICE      =  0.00900
2000::MOUNTVOL_PRICE   =  0.90000
2000::PAGE_PRICE       =  0.01500
2000::PAGEFAULT_PRICE  =  0.00001
2000::PROCESS_PRICE    =  0.09000
2000::QIO_PRICE        =  0.00009
      DISKBLOCK_PRICE  =  0.04000
2000::PRINTJOB_PRICE   =  0.90000
!
```

For more understanding of how resource prices are charged to accounting files, consider the resource prices specified in Figure 20–3.

In Figure 20–3 no price is specified for the DIRECTIO_PRICE keyword, therefore, a price of 0 is applied to all accounting files for direct I/O.

In Figure 20–3 the prefixes specified are NODEA and NODEB. If an accounting file in the CHARGE command has VAX780 for a prefix, the resource prices that do not have prefixes apply to that accounting file. For example, BUFFEREDIO_PRICE would be 0.00010 and CPUSEC_PRICE would be 0.01000.

Figure 20–3 Example Price File

```
TITLE  = 'Resource Charges'
!
! Default Prices
BUFFEREDIO_PRICE =  0.00010
CPUSEC_PRICE     =  0.01000
     .
     .
     .
ELAPSEDSEC_PRICE =  0.00005
FAULTIO_PRICE    =  0.00050
GET_PRICE        =  0.00010
IMAGE_PRICE      =  0.01000
MOUNTVOL_PRICE   =  1.00000
PAGE_PRICE       =  0.02500
PAGEFAULT_PRICE  =  0.00001
PROCESS_PRICE    =  0.10000
QIO_PRICE        =  0.00010
DISKBLOCK_PRICE  =  0.05000
PRINTJOB_PRICE   =  1.00000
!
!Prices for NODEA
!
NODEA::BUFFEREDIO_PRICE =  0.00020
NODEA::CPUSEC_PRICE     =  0.02000
 .
 .
 .
NODEA::ELAPSEDSEC_PRICE =  0.00006
NODEA::FAULTIO_PRICE    =  0.00060
NODEA::GET_PRICE        =  0.00020
NODEA::IMAGE_PRICE      =  0.02000
NODEA::MOUNTVOL_PRICE   =  2.00000
NODEA::PAGE_PRICE       =  0.05000
NODEA::PAGEFAULT_PRICE  =  0.00002
NODEA::PROCESS_PRICE    =  0.20000
NODEA::QIO_PRICE        =  0.00020
       DISKBLOCK_PRICE  =  0.06000
NODEA::PRINTJOB_PRICE   =  2.00000
!
!Prices for NODEB
!
NODEB::BUFFEREDIO_PRICE =  0.00009
NODEB::CPUSEC_PRICE     =  0.00900
```

The following message is displayed:

```
% SPM-W-NOPRICE  Default prices used for PREFIX VAX780
```

Since there is no price specified for direct I/O, a price of 0 is applied.

If an accounting file in the CHARGE command has NODEB for a prefix, the prices specified for NODEB apply to that accounting file and a price of 0 is applied to all other resources. For example, BUFFEREDIO_PRICE would be 0.00009 and CPUSEC_PRICE would be 0.00900. DIRECTIO_PRICE, ELAPSEDSEC_PRICE, and so on would be charged as 0.

20.2.3.3 Specifying Different Prices for Print Pages

With the Charge facility, you can specify different prices for pages printed according to printer queue name and accounting file.

Specify prefixes and printer names according to the following format:

```
prefix::printer_queue_name:PAGE_PRICE = value
```

For example, you may want to charge more per page for documents printed on an LPS40 than for documents printed on a line printer. This charge may vary from system to system as shown in Figure 20–4.

Figure 20–4 Price File Page Prices

```
  .
  .
  .
PAGE_PRICE = 0.03000
!
!  8800 Prices
!
  .
  .
  .
8800::laser_queue:PAGE_PRICE = 0.0500
8800::sys$print_line:PAGE_PRICE  = 0.0200
!
!
!   2000 Prices
!
  .
  .
  .
2000::laser_queue:PAGE_PRICE = 0.0250
2000::sys$print_line:PAGE_PRICE  = 0.0125
```

For more understanding of how page prices are charged to accounting files, consider the page prices specified in Figure 20–4.

If an accounting file has no prefix in the command line, all pages printed are charged at 0.03 per page. If an accounting file has the prefix 8800, the pages from the laser_queue queue are charged at 0.50 and the from the sys$print_line queue are charged at 0.20. All other printer queues in this accounting file will be charged at 0.03.

If the page price keyword or its price is absent from the price file, a page price of 0 is applied to all print queues in all accounting files.

20.2.3.4 Specifying Different Prices for Allocated Disk Blocks

With the Charge facility, you can specify different prices for disks according to volume name by using the following format:

```
volume_name:DISKBLOCK_PRICE = value
```

The following is an example:

```
 SYS_DISK:DISKBLOCK_PRICE =   0.45000
 USER_DISK:DISKBLOCK_PRICE =  0.55000
 BACKUP_DISK:DISKBLOCK_PRICE = 0.50000
```

To report charges for a disk, a disk usage file containing information for that disk must be specified in the CHARGE command line as described in Section 20.3. Section 20.4 describes how to include disk usage information charges in your report.

20.3 Invoking the CHARGE Command

To generate a resource utilization report, give a command of the form:

```
PERFORMANCE REPORT=CHARGE/qualifiers... acct-file[,acct-file,...] -
                                                [charge-file] -
                                                [disk-usage-file]
```

This command accepts three parameters:

- The name of the files containing accounting data
- The name of a price file containing resource prices and report details
- The name of the file containing disk usage information

This command also accepts the qualifiers described in Table 20–4.

Acct-file

The acct-file is a required parameter that specifies the name of the file containing accounting data from the VMS ACCOUNTING utility. One or more accounting files may be specified.

Charge-file

The charge-file specifies the name of the file that contains charges for each billable resource.

To enter prices from the terminal, omit the charge-file parameter from the command line. If the disk-usage parameter is specified, replace the charge-file parameter with two double quotes (" ").

Disk-usage-file

The disk-usage-file specifies the name of a file containing disk usage information from the VMS ANALYZE/DISK_STRUCTURE/USAGE utility. This utility is run for each disk on which charges are to be reported, and the resulting output files are appended into one disk usage file. This parameter is optional and need only be present for disk utilization reporting.

Table 20–4 VAX SPM REPORT=CHARGE Command Qualifiers

Command Qualifier	Description
/BEFORE	Specifies a range of dates for records selected for reporting. The default is /BEFORE=end of file.
/DETAILS	Specifies report details. Default values are the detail keywords specified in the price file (and in their absence, GRAND, ACCOUNTS, and USERS).
/OUTPUT	Specifies a name for the report file. The default is CHARGE.RPT.
/PREFIX	A positional qualifier used after the account-file-spec parameters. This qualifier links a set of resource prices in the charge file with a particular accounting file through a common prefix name.
/SINCE	Specifies a range of dates for records selected for reporting. The default is /SINCE=beginning of file.
/VERSION	Specifies the version number of the utility.

Use the /BEFORE and /SINCE qualifiers to specify the time for which charges are reported.

Keywords for the /DETAILS qualifier are described in Table 20–5.

Table 20–5 Keywords for the /DETAILS Qualifier

Qualifier Keywords	Description
JOBS	Reports on each individual job or process.
JOBTYPES	Reports totals for each unique combination of Account, Username, UIC, and Jobtype. The jobtypes are Batch, Detatched, Interactive, Network, Subprocess, and Print.
UICs	Reports totals for each unique combination of Account and Username.
USERS	Reports totals for each unique combination of Account and Username.
ACCOUNTS	Reports totals for each account.
GRAND	Reports grand totals for each accounting file.
NO_ZEROS_TOTALS	Does not report resource fields with zero unit prices.

The command below generates a report, CHARGE.RPT, for the accounting data in ACCOUNT.DAT. Resource prices and report details are read from the price file SPMCHARGE.TXT.

```
$ PERFORMANCE REPORT=CHARGE ACCOUNT.DAT SPMCHARGE.TXT
```

The command below generates a report, ACCOUNT.CHG, from the accounting data file ACCOUNT.DAT and the price file SPMCHARGE.TXT. Since /DETAILS is given without any items, default details of GRAND, ACCOUNTS, and USERS are reported.

```
$ PERFORMANCE REPORT=CHARGE/OUTPUT=ACCOUNT.CHG -
  EXTRACT.DAT SPMCHARGE.TXT
```

The following command assumes that there are resource prices defined for the prefixes GREEN and YELLOW in the price file CHARGE.DAT:

```
$ PERFORMANCE REPORT=CHARGE ACCT1.DAT/PREFIX=GREEN,ACCT2.DAT -
  /PREFIX=YELLOW, ACCT3.DAT CHARGE.DAT DISK.DAT
```

The following is true about the above command:

- Resource prices associated with the prefix GREEN are applied to the accounting file ACCT1.DAT.
- Resource prices associated with prefix YELLOW are applied to the accounting file ACCT2.DAT.
- Resources prices specified without resource prices in CHARGE.DAT are applied to the accounting file ACCT3.DAT. If there are no such resource prices, a price of 0 is applied to all resources in ACCT3.DAT.
- The disk volume names in CHARGE.DAT have disk usage information in the disk usage file DISK.DAT.

Most of the error conditions that can occur during the generation of a report are reported by standard error messages (see Appendix B of the *VAX SPM Reference Manual*).

20.4 Reporting on Disk Usage Information

To include disk usage information in your Charge report, perform the following steps:

1 Create a disk usage file for the disks on which you wish to report by using the following command:

```
$ ANALYZE/DISK_STRUCTURE/USAGE=[output-file-spec] diskname
```

If no file specification is given, the default output file name of USAGE.DAT is used.

2 Append the disk-usage files into one file using one of the commands shown in the following example:

```
$ APPEND DEVD.DAT,USER1.DAT,SYS_2.DAT DISKUSAGE.DAT
$ SPM$EXAMPLES:SPM$DISK_USAGE.COM
```

3 Specify amounts in the price file for the DISKBLOCK_PRICE keyword for the disks on which you wish to report. The following is an example:

```
DEVD:DISKBLOCK_PRICE = 0.03000
USER1:DISKBLOCK_PRICE = 0.02500
SYS_2:DISKBLOCK_PRICE = 0.03500
```

4 Invoke the CHARGE command and specify the following items:

Accounting file name, price file name, and disk usage file name

At least one of the following keywords for the /DETAILS qualifier:

GRAND
ACCOUNTS
USERS
UICS

The following is an example of a CHARGE command to report disk utilization:

```
$ PERFORMANCE REPORT=CHARGE/DETAILS=(USERS,UICS,ACCOUNTS) ACCOUNT.DAT -
 CHARGE.DAT DISKUSAGE.DAT
```

20.5 Varying Charge Report Contents

This section describes two ways you can vary Charge report contents using VMS ACCOUNTING utility qualifiers. These two ways are: charging different prices for resources based on time, and limiting reports by types, UICs, users, accounts, and so on.

20.5.1 Charging for Resources Based on Time

Follow the instructions below to charge for resources based upon time:

1 Use the VMS ACCOUNTING command with the /SINCE and /BEFORE qualifiers to create accounting files for the time periods for which you want to vary resource prices.

The commands in the following examples create three accounting files: one for 12 a.m. to 7:59 a.m., one for 8:00 a.m. to 5:59 p.m., and the other for 6:00 p.m. to 12 a.m.

```
$ ACCOUNTING /SINCE=14-JAN-1989:00:00/BEFORE=14-JAN-1989:7:59 -
/OUTPUT=MORNING.DAT
$ ACCOUNTING /SINCE=14-JAN-1989:8:00/BEFORE=14-JAN-1989:17:59 -
/OUTPUT=DAY.DAT
$ ACCOUNTING /SINCE=14-JAN-1989:18:00/BEFORE=14-JAN-1989:00:00 -
/OUTPUT=EVENING.DAT
```

2 Specify prefixes and resource prices in the price file for the three time periods, as shown in the following price file segment.

```
! Morning Prices
MORN::BUFFEREDIO_PRICE = 0.0009
MORN::CPUSEC_PRICE = 0.00999
MORN::DIRECTIO_PRICE = 0.00049
  .
  .
  .
! Daytime Prices
DAY::BUFFEREDIO_PRICE = 0.00030
DAY::CPUSEC_PRICE = 0.03000
DAY::DIRECTIO_PRICE = 0.00080
  .
  .
  .
! Evening Prices
EVENING::BUFFEREDIO_PRICE = 0.00010
EVENING::CPUSEC_PRICE = 0.01000
EVENING::DIRECTIO_PRICE = 0.00050
  .
  .
  .
```

3 Specify prefixes for each accounting file in the CHARGE command line, as in the following example:

```
$ PERFORMANCE REPORT=CHARGE MORNING.DAT/PREFIX=MORN,DAY.DAT/PREFIX=DAY
EVENING.DAT/PREFIX=EVENING CHARGE.DAT
```

20.5.2 Reporting on Specific Accounting File Components

Use the following VMS ACCOUNTING qualifiers to create an accounting file of specific records:

- /TYPE=(print,process)
- /OWNER
- /PROCESS, /UICS, /USERS and /ACCOUNTS

Refer to the *VAX/VMS Accounting Utility Manual* for more information.

21 Using VAX SPM on a VAXcluster System

This chapter provides the following information about using VAX SPM to collect, archive, and evaluate performance on your VAXcluster system:

- Cluster-specific collection information
- Cluster-specific archive information
- Discussion of balancing the workload on a VAXcluster system
- Discussion of isolating disk contention on a VAXcluster system

21.1 Cluster-Specific Collection Information

Chapter 3 describes the PERFORMANCE COLLECT command and how to begin data collection using VAX SPM. This section provides the following cluster-specific collection information to supplement the information in Chapter 3:

- VAXcluster system mode collections
- VAXcluster system disk name conventions
- Node-specific initialization commands

21.1.1 VAXcluster System Mode Collections

This section describes invoking collections from a node within a VAXcluster system. To invoke VAX SPM on or from a remote node, read Chapter 22.

Begin cluster-wide collections by giving the PERFORMANCE COLLECT command on any node and specifying one of the following for the node name parameter:

- An asterisk (*)
- A list of node names
- A node name

The following command begins collections on all nodes in a VAXcluster system:

```
$ PERFORMANCE COLLECT=TUNE *
```

The following command begins collections on nodes GREEN, BLUE, and AQUA:

```
$ PERFORMANCE COLLECT=TUNE GREEN,BLUE,AQUA
```

The following command begins collections on node YELLOW:

```
$ PERFORMANCE COLLECT=TUNE YELLOW
```

If the above command is given on node YELLOW, collection begins only for node YELLOW. Although collection is taking place on only one node, specifying the node name parameter invokes VAX SPM in VAXcluster system mode.

There are three requirements for running VAX SPM in VAXcluster system mode:

- Each node must be a member of the local VAXcluster.
- DECnet must be installed on each node.
- A standard or collector version of VAX SPM must be installed on each node.

In VAXcluster system mode, a process runs the SPM$REMOTE object on the remote node in the SPM account. (The SPM account is an account established when VAX SPM is installed.) This process oversees the translation of logical names and the execution of the PERFORMANCE command on the remote node.

Cluster collections can be started using either the PERFORMANCE COLLECT=TUNE or the PERFORMANCE COLLECT=CAPACITY command. If the PERFORMANCE COLLECT=TUNE command is invoked, a detached process called SPM_TUNE is created on each node for which the collection is started. Similarly, if the PERFORMANCE COLLECT=CAPACITY command is invoked, a detached process called SPM_CAPACITY is created on each node.

As with single-node collection, you may specify qualifiers such as /START, /STOP, /INQUIRE, and /NEW_FILE to control data collections across a cluster. Use one of these qualifiers and the node name parameter to stop or inquire about a collection, or to initiate a new file operation for every node specified.

When collecting cluster data for archiving into a history file, it is best to keep collections consistent for all nodes. The PERFORMANCE collection command, however, provides two methods of varying collections for each node in a cluster. One method is through the use of node-specific initialization commands, which are described in Section 21.1.3. The other method is through the use of global and local qualifiers in the COLLECT command.

Collection qualifiers can be placed globally to apply to all nodes in the command line or with a node name (locally) to apply only to that node. If given globally, qualifiers apply to data collection for every node specified or implied. Consider the following example:

```
$ PERFORMANCE COLLECT=TUNE/START/INTERVAL=300 -
GREEN/OUTPUT=GREEN.DAT,BLUE/INTERVAL=500/OUTPUT=BLUE.DAT
```

Qualifiers in the preceding command affect collections on each node in the following way:

- Tuning data collection is started on GREEN with a collection interval of 300 seconds (global qualifier).
- The output file for node GREEN is GREEN.DAT (local qualifier).

- Tuning data collection is started on node BLUE with a collection interval of 500 seconds (local qualifier).
- The output file for node BLUE is BLUE.DAT (local qualifier).

If /OUTPUT is omitted, a log file for each specified node is written in the SPM account. The log file names are SPM$COLLECT_TUNE_node-name.DAT when the PERFORMANCE COLLECT=TUNE is invoked, and SPM$COLLECT_CAPACITY_node-name.DAT when the PERFORMANCE COLLECT=CAPACITY command is invoked.

In the following command, PERFORMANCE COLLECT=TUNE is used and the /OUTPUT qualifier is omitted:

```
$ PERFORMANCE COLLECT=TUNE GREEN,BLUE
```

Since /OUTPUT is omitted, the log file names are SPM$COLLECT_TUNE_GREEN.DAT and SPM$COLLECT_TUNE_BLUE.DAT

When data collection is invoked in cluster mode, SPM messages are preceded by additional messages of the form:

```
%SPM-I-NODEMSG, "operation" message for node "node-name"
```

In these messages, the word “operation” is replaced by the command action (for example, STOP, START, INQUIRE) and the word “node-name” is replaced by the specific node name.

For example, if you attempt to start tuning data collection in cluster mode on node GREEN , and tuning data collection is already in progress on that node, the following messages appear:

```
$PERFORMANCE COLLECT=TUNE GREEN

%SPM-I-NODEMSG, START message for node GREEN
    %SPM-E-COALRU, Collections already running
```

21.1.2 VAXcluster Disk Name Conventions

In an environment that includes VAXclusters, HSC style controllers, and MSCP-served disks, certain rules govern the naming of disk devices. Each node in a VAXcluster sees a variety of local and remote disks as well as controllers. Certain disks on remote nodes will not be reachable from other nodes. The rules that govern what disks and controllers are seen from a particular node, and what names are assigned them, are as follows:

- If no allocation class is assigned for the local node, the local node sees local devices as local-node-name$disk-name.
- If an allocation class L is assigned for the local node, the local node sees local devices as $L$disk-name.
- If no allocation class is assigned for the remote node, the local node sees remote devices as remote-node-name$disk-name.
- If an allocation class R is assigned for the remote node, the local node sees devices on the remote node as $R$disk-name.

- If two nodes have a drive with the same name, and both nodes have the same allocation class value, then the drive must be dual-ported between the two nodes.
- Disks on a remote node that are not MSCP-served (or not on an HSC) cannot be seen from the local node.
- Controllers on the local node are seen from the local node by their names (for example, MBAn for MASSBUS controllers, PUAn for UDA50 controllers on the UNIBUS, and DMAn for other UNIBUS controllers).
- Controllers on remote nodes are seen from the local node as remote-node-name$, whether or not an allocation class is assigned.

Figure 21–1 illustrates these rules as applied to an example VAXcluster consisting of three VAX processors and an HSC50. The node names are A, B, C, and D (the HSC50). DR and DB are MASSBUS disks, while DU and DJ are DSA disks that can be controlled by a UDA50 or the HSC50. Some of the disks are MSCP-served and others are purely local to their node. One of the MASSBUS disks is dual-ported between nodes A and B. The names of the disks and controllers that can be seen from each of the three VAX processors (nodes A, B, and C) are shown in this figure.

By using an allocation class, a disk that fails over to a secondary controller retains its name. For example, node C can access $7$DBA3 either through node A or node B, but in either case this disk is seen as $7$DBA3. Disk names in SPM reports remain the same when failover occurs, but new server names may appear under server statistics. If a disk or controller can no longer be reached, its statistics will be recorded as zeroes.

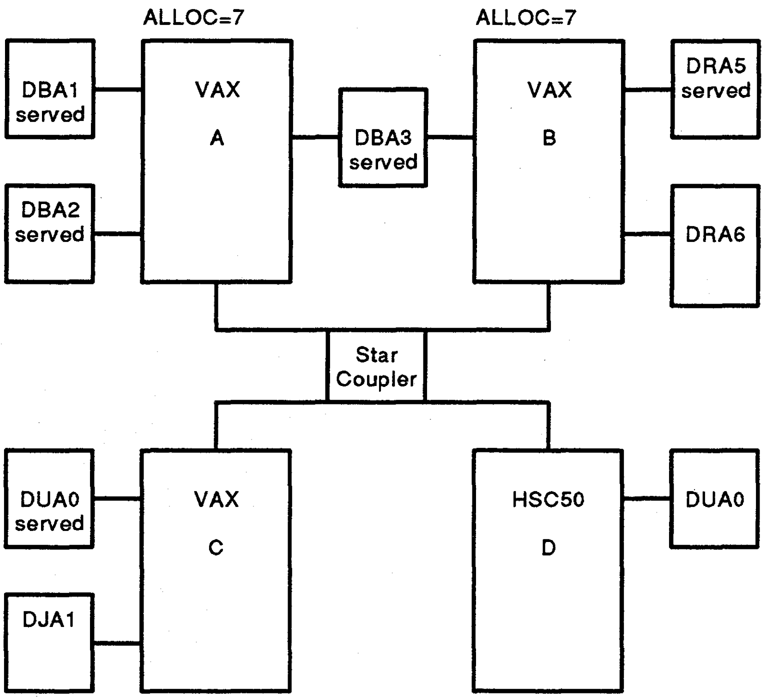
21.1.3 Using Node-Specific Initialization Commands

You can begin cluster collections with one command and have each node execute a command customized to its individual requirements. For example, one node may collect DECnet devices called XEA1 at one collection interval, while another collects DECnet devices called ETA1 at a different collection interval.

This is accomplished through the use of node-specific initialization command lines. An initialization command line has the following characteristics:

- It is referenced by the logical SPM$INI_COLLECT_TUNE for the COLLECT=TUNE command or the logical SPM$INI_COLLECT=CAPACITY for the COLLECT=CAPACITY command.
- It may be either a command line or a file containing a command line
- It is a valid PERFORMANCE command and it is not preceded by a $.

Figure 21–1 Disk and Controller Name Conventions



	Node A	Node B	Node C
Disks	$7$DBA1	$7$DBA1	$7$DBA1
seen	$7$DBA2	$7$DBA2	$7$DBA2
	$7$DBA3	$7$DBA3	$7$DBA3
	$7$DRA5	$7$DRA5	$7$DRA5
		$7$DRA6	
	C$DUA0	C$DUA0	C$DUA0
			C$DJA1
	D$DUA0	D$DUA0	D$DUA0
Servers	MBAn	MBAn	PUAn
seen		A$	A$
	B$		B$
	C$	C$	
	D$	D$	D$

When a command is given to begin cluster collections, the command applied to an individual node is determined by the interactive command line and the initialization command line together, according to the following rules:

- The content of the node-specific initialization command line is applied first and the content of the interactive command line is processed last. This means that qualifiers in the interactive command line can override qualifiers in the initialization command line.
- If a qualifier that appears in the node-specific initialization command line is not explicitly negated or changed in the interactive command line, the qualifier is used in the final command executed on the node.

- If no log file name is specified on either the interactive or the initialization command lines, a default log file name is used. Depending on the collection command, the default log file name is SPM$COLLECT_TUNE_node-name.DAT or SPM$COLLECT_CAPACITY_node-name.DAT.
- The logical names SPM$INI_COLLECT_TUNE or SPM$INI_COLLECT_CAPACITY are treated as search lists, but elements within them are treated as ordinary logical names. See the *VAX/VMS DCL Concepts Manual* for more information on search lists.
- All logical names are translated in the context of the specific node for which data collection is to be started. Therefore, logical names in the interactive command may have the same or different translations on each node. To avoid translation failures in a VAXcluster, use logical names on the interactive command line that have the same meaning across the cluster. If logical names do not have the same meaning for all nodes, specify them in node-specific initialization commands.

For example, the following commands show how qualifiers in the interactive command can override qualifiers in the initialization command. The interactive command line is:

```
$ PERFORMANCE COLLECT=TUNE/CLASS=(NODISK)
```

The SPM$INI_COLLECT_TUNE logical name translates to:

```
 PERFORMANCE COLLECT=TUNE/CLASS=(DISK,DEVICE,NOLOCK,SCHEDULER) -
 /DISK=GREEN$DBA1/DEVICE=XMA0
```

The following command is executed on the local node:

```
$ PERFORMANCE COLLECT=TUNE/CLASS=(NODISK,DEVICE,NOLOCK,SCHEDULER)
```

The /NODISK keyword in the interactive command line overrides the DISK keyword in the initialization command line.

In the example below, the /OUTPUT qualifier is specified in the node-specific initialization command, but not in the interactive command.

The interactive command line is:

```
$ PERFORMANCE COLLECT=TUNE/START *
```

The SPM$INI_COLLECT_TUNE logical name on node GREEN translates to:

```
  PERFORMANCE COLLECT=TUNE/START/OUTPUT=1985FEB02.DAT
```

The resulting command string is executed on node GREEN:

```
$ PERFORMANCE COLLECT=TUNE/START/OUTPUT=1985FEB02.DAT
```

Since the /OUTPUT qualifier is not negated or overridden by the interactive command, it appears in the resultant command string for the node.

In the next example, logical names are provided in the interactive command line.

The following interactive command line starts data collections on a 2-node cluster comprised of nodes GREEN and BLUE.

```
$ PERFORMANCE COLLECT=TUNE/START/OUTPUT=SPM$DISK:1985FEB14.DAT *
```

The logical name SPM$DISK is translated in the context of node GREEN for beginning data collection on GREEN, and in the context of node BLUE for beginning data collection on BLUE. The logical name may have the same or different translations on each node.

21.2 Creating a Cluster-Wide History File

Data from log files created by different VAXcluster system nodes can be archived into the same history file. This allows you to catalog, by node and time, cluster-wide performance data in a single file for immediate as well as long-term trend analysis.

Chapter 15 describes the PERFORMANCE ARCHIVE command and how to create a history file using VAX SPM. This section provides the following cluster-specific information to supplement the information in Chapter 15:

- Using node names in a cluster-wide history file
- Specifying a node name
- Supplying allocation class information
- Listing information in a cluster-wide history file
- Deleting records from a cluster-wide history file

21.2.1 Using Node Names in a Cluster-Wide History File

VAX SPM maintains the integrity of a history file by verifying the validity of data before it is added. Incoming data is verified using a table of node names, which is maintained in the history file. When new data is added to the history file, the node name of new data is checked against the table to ensure the following:

- A log file with a blank node name is not added to a history file containing named nodes.
- A log file with a named node is not added to a history file containing a blank node name.

The SCS node name is set by the value of the system parameter SCSNODE. If this value was not set using the system parameter before collecting data, it is zero or blank in the log file.

Use the /NODE_NAME qualifier with the PERFORMANCE ARCHIVE=HISTORY command to archive log files that lack node names. /NODE_NAME can supplement or override information recorded in the input log file. For example, to add a log file from a node with no name to a history file containing nodes with names, use /NODE_NAME to supply a name.

If you create a history file for a node with no name and that node later becomes a member of a VAXcluster system, you will have to create a new history file (from log files) and supply a node name for the old data in order to catalog other named nodes in the same history file.

21.2.2 Specifying a Node Name

Use the /NODE_NAME qualifier to supplement or override node name information in an input log file that is being archived (/ADD or /REPLACE operation). You can also use the /NODE_NAME qualifier to select records for a specific node during /LIST and /DELETE operations. A blank node name may be supplied to override a node name in the log file. If this qualifier is omitted from the command line, then what was recorded in the log file is used. This qualifier may be specified globally, affecting all log files, or locally on a per log file basis.

In the example below, /NODE_NAME=MAROON is used globally and specifies the node name for log files NONODE*.DAT and NOCOLOR*.DAT, while /NODE_NAME=PURPLE is used locally and specifies the node name for log file GREEN.DAT.

```
$ PERFORMANCE ARCHIVE=HISTORY/NODE_NAME=MAROON HISTORY.DAT -
  NONODE*.DAT,GREEN.DAT/NODE_NAME=PURPLE,NOCOLOR*.DAT
```

The node name specified with /NODE_NAME can also prefix any disk or device name from the input log file that is lacking a node name prefix. For example, if /NODE_NAME=AQUA were specified and a disk name of DBA0 were found in the input log file, then the disk would be renamed during archiving to be AQUA$DBA0. This is useful for archiving log file data from a noncluster VAX into the history file of a machine that is now part of a cluster. Note that if allocation class information is supplied for the specified node using the /ALLOCATION_CLASS qualifier, then the prefix will be changed to be the allocation class.

In this example, regardless of the node name contained in the log file SPM$COLLECT_CAPACITY.DAT, data is archived under the name MAROON.

```
$ PERFORMANCE ARCHIVE=HISTORY HISTORY.DAT -
  SPM$COLLECT_CAPACITY.DAT/NODE_NAME=MAROON
```

21.2.3 Supplying Allocation Class Information

A disk spindle may be served by either of two HSC controllers or VAX nodes. In order to report on this disk spindle over time, its name must remain constant regardless of the server hosting it. To accomplish this, the same allocation class is assigned to HSC controllers or VAX nodes that potentially serve the same disk. The allocation class instead of the node name is then prefixed to the disk name and does not change in case of failover. For example, if the allocation class is 7, disk DUA0 would be known as $7$DUA0 no matter which controller or node was actually serving the disk.

Versions of SPM prior to 2.2 did not record the server-independent name of such dual-ported disk spindles. Therefore, these disks appear in log files with node-specific names such as MAROON$DUA0. Data collected on a system prior to its becoming a member of a VAXcluster system may also be missing allocation class information. In both cases, a mechanism is needed to allow the specification of the missing allocation class information when archiving these types of log files.

The /ALLOCATION_CLASS qualifier is used to supply missing allocation class information. This qualifier can also be used to override the node prefix of any disk or device in the input log file(s), which is prefixed by a node name declared using this qualifier. The format of this qualifier is:

```
    /ALLOCATION_CLASS=(class:name,...)
```

"Class" is the allocation class number to be associated with the node name specified by "name." There are no defaults for this qualifier. This is a global qualifier, applying equally to data found in all input log files specified on the command line.

When a disk device name is prefixed by a node name specified by the /ALLOCATION_CLASS qualifier, the prefix of the disk name is modified to contain the associated allocation class value.

If the name of the disk device in the input log file is YELLOW$DUA0, the following command would cause the disk to be archived as $255$DUA0:

```
$ PERFORMANCE ARCHIVE=HISTORY/ALLOCATION_CLASS=(255:YELLOW)...
```

More than one node name may be associated with an allocation class value. The command below specifies that all disk and device names found in the input log file(s) prefixed by either MAROON$ or AQUA$ would have their prefix changed to $255$. Disks and devices prefixed by PURPLE would have their prefix changed to $7$.

```
/ALLOCATION_CLASS=(255:MAROON,255:AQUA,7:PURPLE)
```

With the /NODE_NAME (Section 21.2.2) qualifier also present in the command line, disk or device name alteration occurs first using the value specified by /NODE_NAME, then using the value specified by /ALLOCATION_CLASS. The /NODE-NAME qualifier is used together with the /ALLOCATION_CLASS qualifier when disk names in the log file have no node prefix (for example, DUA0).

If the input log files lack a node name and include data for a disk DUA1, then the following command line archives the disk to the history file as $10$DUA1:

```
$ PERFORMANCE ARCHIVE=HISTORY/NODE_NAME=MAROON -
 /ALLOCATION_CLASS=(10:MAROON) COLOR_CLUSTER.DAT MAROON_MAY.DAT
```

21.2.4 Listing Information in a Cluster-Wide History File

Use the /LIST qualifier to list information in a history file that contains data from more than one node in a VAXcluster system. There are two keywords for the /LIST qualifier: BRIEF and FULL. With BRIEF, the listing includes the history file archive interval, the node name and SCS system ID, and the begin/end times of continuous data in the history file. With FULL, the listing includes additional information about the classes of statistics archived. No server, disk, and device names are used as in the case of a single-node history file, unless the /NODE_NAME (Section 21.2.2) qualifier is used to specify a single-node name.

In the following example, the FULL keyword with the /LIST qualifier displays the history time interval, system name and ID, periods for which continuous data was found, and the classes of statistics archived. Note that data for node MAROON was not present for the period 00 a.m. to 10 a.m. on April 2, 1988. To obtain disk, device, and server data for node AQUA, use /NODE_NAME=AQUA in the command.

```
$ PERFORMANCE ARCHIVE=HISTORY/LIST=FULL HISTORY.DAT
Data was found in history file.
History file interval is 15 minutes.
Node(s):             System id hi:  System id lo:
   AQUA                          9           3333
   MAROON                       9           4444
From :  1-APR-1988 00:00:00.00  To :  2-APR-1988 00:00:00.00
Nodes:                Archived data:
   AQUA                  mem,cpu,page,io,xqp,dsk,dev,fcp,lck,scs
   MAROON               mem,cpu,page,io,xqp,dsk,dev,fcp,lck,scs
From :  2-APR-1988 00:00:00.00  To :  2-APR-1988 10:00:00.00
Nodes:                Archived data:
   AQUA                  mem,cpu,page,io,xqp,dsk,dev,fcp,lck,scs
From :  2-APR-1988 10:00:00.00  To :  2-APR-1988 15:15:00.00
Nodes:                Archived data:
   AQUA                  mem,cpu,page,io,xqp,dsk,dev,fcp,lck,scs
   MAROON               mem,cpu,page,io,xqp,dsk,dev,fcp,lck,scs
```

21.2.5 Deleting Cluster-Wide Records

Use the /DELETE qualifier (Section 15.5) with the /NODE_NAME qualifier (Section 21.2.2) to delete records for a specific node, and with the /BEFORE and /SINCE qualifiers (Section 15.2.5) to select records for deletion based on time.

The following command deletes records from the history file for node MAROON between the specified /SINCE and /BEFORE times:

```
$ PERFORMANCE ARCHIVE=HISTORY/DELETE/NODE_NAME=MAROON -
  /SINCE=14-FEB-1985:08:00/BEFORE=14-FEB-1985:10:00 V30HISTORY.DAT
```

If the /DELETE command is not bound by a /NODE_NAME qualifier, then all data for the specified period is deleted.

Note that deleting all data associated with a particular node in the history file will not delete that node's identification record from the file. Therefore, the only way to remove all traces of a node in the history file is to recreate the file.

The command below deletes all records in the history file for node PURPLE; however, the node name remains in the node name table.

```
$ PERFORMANCE ARCHIVE=HISTORY/DELETE/NODE_NAME=PURPLE
```

21.3 Workload Balancing on a VAXcluster System

Workload balancing is a way to achieve maximum resource utilization on a VAXcluster system. It usually begins after individual nodes have been tuned.

In the simplest case, workload balancing is achieved by moving processes from one node to another until all nodes share the work and resources evenly. This achieves a balanced workload if all nodes are the same CPU type and all users are doing equal work. Since this is not always the case, one must consider a number of factors when attempting to balance a cluster workload. For example, since all CPUs are not equal, it may be better for larger CPUs to carry a heavier workload than smaller CPUs.

This section does not address all the factors involved in workload balancing. See the *Guide to VAX/VMS Performance Management* for more information. This section describes how to get started, and how to use VAX SPM range graphs and cluster tabular reports to evaluate cluster workloads.

Perform the following steps to collect and archive cluster data and generate Process Count, CPU, and Memory Range graphs, and a By Node Tabular report.

1 If you do not have a history file of data, collect a week's worth of data using a command similar to the following:

```
$ PERFORMANCE COLLECT=CAPACITY/CLASS=(PROCESS,FILE_PRIMITIVES) -
  /INTERVAL=300/START=8:00/END=18:00 *
```

This command collects the default statistics (CPU, DISK, IO, Memory, XQP, Pagefault), standard process metrics, and FCP information for every 5 minutes from 8 a.m. to 6 p.m. on all nodes in a VAXcluster system.

2 Create a history file of the data using the following command:

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE/INTERVAL=300 HISTORY.DAT -
SPM$COLLECT_TUNE*.DAT
```

This command archives CPU, memory IO, page fault, XQP, and disk data.

3 Type the following command to generate cluster CPU, Memory, and Cluster Process Count Range graphs, and a By Node Tabular report:

```
$ PERFORMANCE REPORT=HISTORY HISTORY.DAT/CLUSTER -
/GRAPH=(CPU,PROCESS,MEMORY)/STYLE=RANGE/SHIFT=PRIME/WEEKLY/AVERAGE -
/TABULAR=BYNODE/BEFORE="26-APR-1987"/SINCE="29-MAR-1987" -
/OUTPUT=REPORT.RPT
```

This command reports prime hours of the average week from a month's worth of data. Although this provides a good representation of system utilization, you could report on less data (for example, an average day over the course of a week).

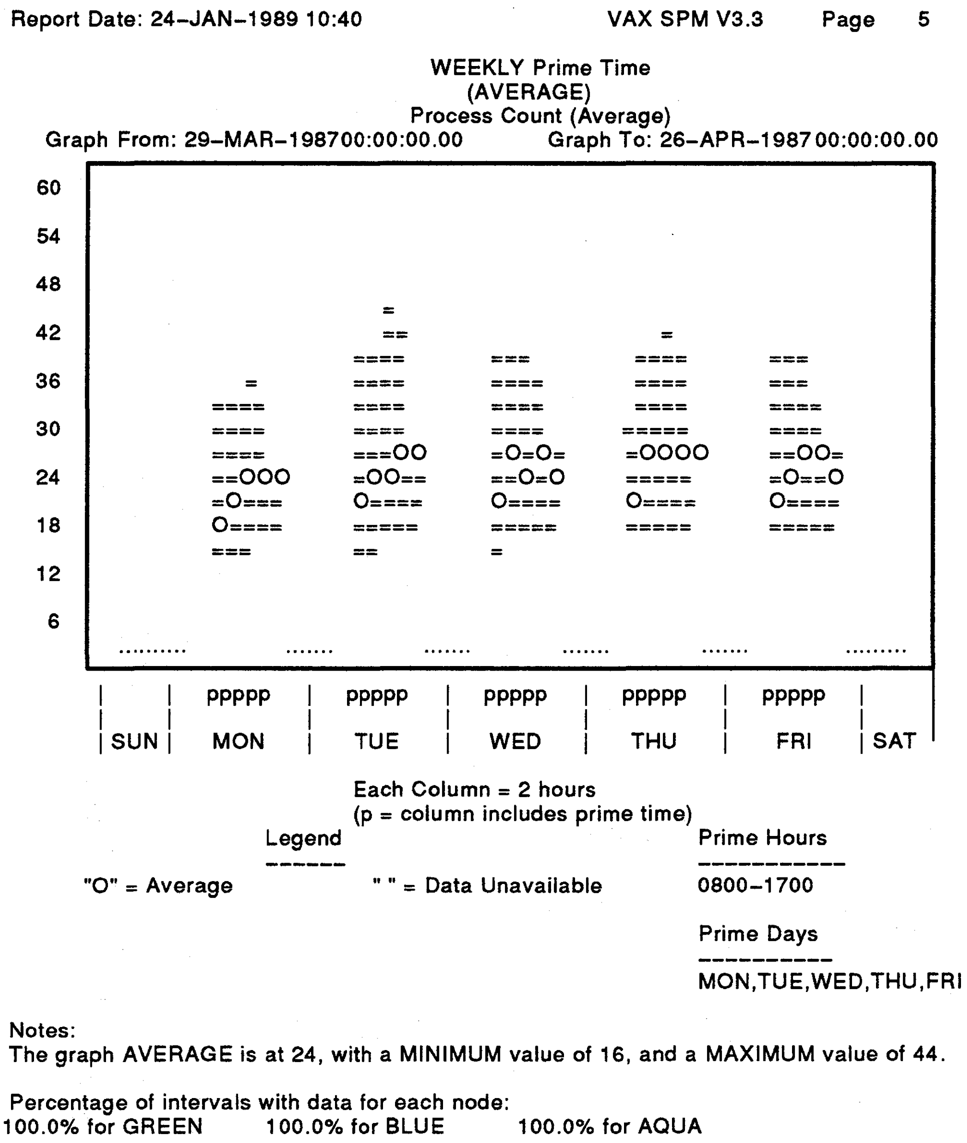
The Cluster Process Count Range graph generated by this command shows the maximum, minimum, and average number of processes on nodes in a cluster. This graph helps you determine if the nodes in a cluster have an equal number of users. Figure 21–2 is an example of a Cluster Process Count Range graph.

In Figure 21–2, the highest number of process counts is between 36-46, the average number is between 18-24, and the lowest is between 16-18.

The large difference between the maximum and minimum may indicate that there are more users on one node than on another. If this is the case, moving processes from the node with the most users to other nodes with fewer users could balance the cluster workload. Unless all users are doing the same type of work and all CPUs are of the same type, however, CPU and memory utilization on these nodes must also be considered. A well-balanced workload cannot be achieved by balancing the number of users across nodes if this causes an imbalance in cluster resource utilization.

Figure 21–3 is an example of a CPU Usage Range graph. It shows that the maximum amount of CPU used is 90-100% and the minimum amount of CPU used about 45%. This may indicate that moving processes using a lot of CPU from the busiest node to other nodes could create a more balanced cluster workload.

Figure 21–2 Cluster Process Count Range Graph



```
Report Date: 24-JAN-1989 10:40                        VAX SPM V3.3      Page    5

                                        WEEKLY Prime Time
                                           (AVERAGE)
                                     Process Count (Average)
   Graph From: 29-MAR-198700:00:00.00           Graph To: 26-APR-198700:00:00.00

60

54

48
                          =
42                       ==                   =
                       ====      ===       ====      ===
36              =      ====      ====      ====      ===
             ====      ====      ====      ====      ====
30           ====      ====      ====     =====      ====
             ====      ===OO     =O=O=     =OOOO     ==OO=
24           ==OOO     =OO==     ==O=O     =====     =O==O
             =O===     O====     O====     O====     O====
18           O====     =====     =====     =====     =====
             ===       ==        =
12

 6
       ..........    .......    .......    .......    .......    .........

     |   |  ppppp  |  ppppp  |  ppppp  |  ppppp  |  ppppp  |    |
     |   |         |         |         |         |         |    |
     |SUN|   MON   |   TUE   |   WED   |   THU   |   FRI   |SAT |

                           Each Column = 2 hours
                           (p = column includes prime time)
                  Legend                                Prime Hours
                  ------                                -----------
  "O" = Average             " " = Data Unavailable      0800-1700

                                                        Prime Days
                                                        ----------
                                                        MON,TUE,WED,THU,FRI

Notes:
The graph AVERAGE is at 24, with a MINIMUM value of 16, and a MAXIMUM value of 44.

Percentage of intervals with data for each node:
100.0% for GREEN      100.0% for BLUE      100.0% for AQUA
```

Figure 21–3 Cluster CPU Usage Range Graph

```
Report Date: 24-JAN-1989    10:40                      VAX SPM V3.3       Page    3
                                  WEEKLY Prime Time
                                      (AVERAGE)
                                 CPU Utilization (Percent)
Graph From: 29-MAR-1987   00:00:00.00           Graph To: 26-APR-1987   00:00:00.00
100%             =  ==       ====                   =====        =   =
                 =====       =====       === =      =====        == ==
 90%             =====       =====       =====      =O===        ====O
                 =====       =====       =O==O      O===O        =O===
 80%             ====O       ====O       ==O==      ==O==        O=O==
                 =====       O==O=       O==O       ===O=        ===O
 70%             ===O=       =OO==       ====       =====        ====
                 ==O=        =====       ====       ===          =  =
 60%             O==         ====         ===       = =          =
                 =O=         ===          ===          =
 50%             ==          = =             =
                 =           =
 40%

 30%

 20%

 10%
         ...........    .......     .......     .......     .......     .........

       |      |   ppppp   |   ppppp   |   ppppp   |   ppppp   |   ppppp   |     |
       |      |           |           |           |           |           |     |
       | SUN  |    MON    |    TUE    |    WED    |    THU    |    FRI    | SAT |

                                Each Column = 2 hours
                            (p = column includes prime time)
              Legend                                              Prime Hours
              ------                                              -----------
       "O" = Average                 " " = Data Unavailable       0800-1700
                                                                  Prime Days
                                                                  ----------
                                                                  MON,TUE,WED,THU,FRI
         Notes:
         The graph AVERAGE is at 75%, with a MINIMUM value of 45% and a MAXIMUM
         value of 100%.
         Percentage of intervals with data for each node:
       100.0% for GREEN          100.0% for BLUE      100.0% for AQUA
```

Use the Cluster By Node Tabular report to determine which nodes are busy and which nodes are not. Figure 21–4 is an example and shows how each node on the VAXcluster system is utilizing CPU. In Figure 21–4, node GREEN is busy 95.2 % of the time, node BLUE 76.3%, and node AQUA 61.9%. A logical next step might be to examine VAX SPM Process Metrics reports as described in Chapter 8 for each node to determine the processes that are using the most CPU. You may achieve a more balanced workload by moving these processes to less busy nodes.

In addition to CPU utilization, memory utilization by nodes in a cluster is another factor to consider when balancing the workload. A Memory Utilization Range graph shows maximum, minimum, and average memory utilization for a cluster.

In Figure 21–5, the highest amount of memory used is about 55-60%, and the lowest amount of memory used is 34.8%. This indicates that memory utilization is evenly balanced across the cluster and that memory is not heavily utilized.

The memory utilization for each node is shown in Figure 21–5. Node GREEN is 43.7% utilized, node BLUE is 47.3% utilized, and node AQUA is 46.7% utilized. If the memory utilization were unbalanced across the cluster, the BY NODE memory statistics shown in Figure 21–4 would be useful in determining which nodes may be under- or overutilized. Balancing the workloads on these nodes may create a more balanced cluster workload.

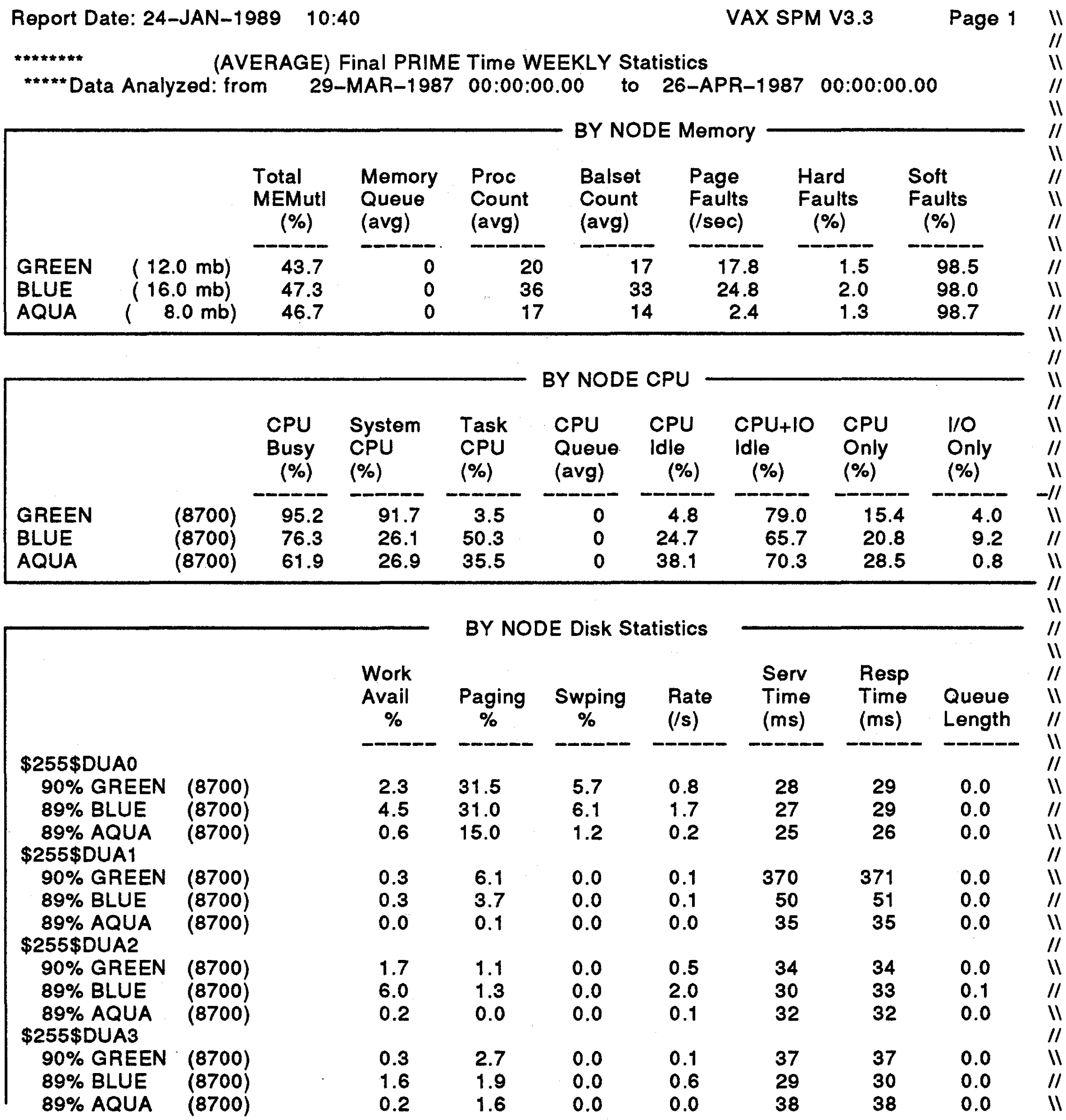
21.4 Isolating Disk Contention in a VAXcluster System

Section 8.3 describes how to isolate the causes of an I/O limitation on a single node. If you have investigated I/O on each node in a cluster and an I/O problem persists, the next step is to evaluate I/O for the cluster.

Disk contention is one cause of an I/O limitation that requires evaluation from a cluster perspective. It occurs when one node's I/Os to a disk are getting blocked by I/Os from other nodes. Disk contention shows up on a single node disk statistics report as a disk having a high response time and a low rate per second.

Begin evaluating disk contention by combining the log files from all nodes in the cluster into a history file and generating a report to identify the most active disks. There are a number of ways you can evaluate the activity on these disks. One way is to locate the most active files on these disks using the DISPLAY=FILES command described in Chapter 9. Another way is to generate a By Cluster Tabular report to learn each node's contribution to the high I/O rate on a disk.

Figure 21–4 Cluster By Node Tabular Report



```
Report Date: 24-JAN-1989   10:40                                VAX SPM V3.3      Page 1   \\
                                                                                           //
*********        (AVERAGE) Final PRIME Time WEEKLY Statistics                              \\
 *****Data Analyzed: from     29-MAR-1987  00:00:00.00    to  26-APR-1987  00:00:00.00     //
                                                                                           \\
 ------------------------------------------ BY NODE Memory ------------------------------  //
                                                                                           \\
                          Total    Memory    Proc     Balset    Page     Hard     Soft     //
                          MEMutl   Queue     Count    Count     Faults   Faults   Faults   \\
                           (%)     (avg)     (avg)    (avg)     (/sec)    (%)      (%)     //
                          ------   ------    ------   ------    ------   ------   ------   \\
 GREEN    ( 12.0 mb)       43.7        0        20        17      17.8      1.5     98.5   //
 BLUE     ( 16.0 mb)       47.3        0        36        33      24.8      2.0     98.0   \\
 AQUA     (   8.0 mb)      46.7        0        17        14       2.4      1.3     98.7   //
 ----------------------------------------------------------------------------------------  \\
                                                                                           //
 ---------------------------------------- BY NODE CPU ----------------------------------   \\
                                                                                           //
                            CPU    System   Task     CPU     CPU     CPU+IO   CPU     I/O   \\
                            Busy   CPU      CPU      Queue   Idle    Idle     Only    Only  //
                            (%)    (%)      (%)      (avg)    (%)     (%)     (%)     (%)   \\
                          ------  ------   ------   ------  ------  ------   ------  ------ -//
 GREEN          (8700)      95.2    91.7     3.5        0     4.8     79.0     15.4     4.0  \\
 BLUE           (8700)      76.3    26.1    50.3        0    24.7     65.7     20.8     9.2  //
 AQUA           (8700)      61.9    26.9    35.5        0    38.1     70.3     28.5     0.8  \\
 ------------------------------------------------------------------------------------------ //
                                                                                            \\
 ------------------------------------ BY NODE Disk Statistics -------------------------     //
                                                                                            \\
                                Work                                Serv    Resp            //
                                Avail   Paging   Swping   Rate      Time    Time    Queue   \\
                                 %        %        %      (/s)      (ms)    (ms)    Length  //
                                ------  ------   ------  ------   ------  ------  ------    \\
 $255$DUA0                                                                                  //
   90% GREEN   (8700)            2.3    31.5      5.7     0.8        28      29     0.0     \\
   89% BLUE    (8700)            4.5    31.0      6.1     1.7        27      29     0.0     //
   89% AQUA    (8700)            0.6    15.0      1.2     0.2        25      26     0.0     \\
 $255$DUA1                                                                                  //
   90% GREEN   (8700)            0.3     6.1      0.0     0.1       370     371     0.0     \\
   89% BLUE    (8700)            0.3     3.7      0.0     0.1        50      51     0.0     //
   89% AQUA    (8700)            0.0     0.1      0.0     0.0        35      35     0.0     \\
 $255$DUA2                                                                                  //
   90% GREEN   (8700)            1.7     1.1      0.0     0.5        34      34     0.0     \\
   89% BLUE    (8700)            6.0     1.3      0.0     2.0        30      33     0.1     //
   89% AQUA    (8700)            0.2     0.0      0.0     0.1        32      32     0.0     \\
 $255$DUA3                                                                                  //
   90% GREEN   (8700)            0.3     2.7      0.0     0.1        37      37     0.0     \\
   89% BLUE    (8700)            1.6     1.9      0.0     0.6        29      30     0.0     //
   89% AQUA    (8700)            0.2     1.6      0.0     0.0        38      38     0.0     \\
```

Figure 21–5 Cluster Memory Usage Range Graph

```
Report Date: 24-JAN-1989  10:40                          VAX SPM V3.3      Page    4
                                   WEEKLY Prime Time
                                       (AVERAGE)
                                Memory Utilization (Percent)
   Graph From: 29-MAR-1987 00:00:00.00          Graph To: 26-APR-1987 00:00:00.00
     +------------------------------------------------------------------------------------+
100% |
 90% |
 80% |
 70% |
 60% |                                                       =
     |                            ==            =            =
 50% |               ==         ==OO          O=O         =OOOO         = O ^
     |            =OOOO        =OO==         ==O=O        O ===        =OO=O
 40% |            O====        O====         O            =            O
     |            =
 30% |
 20% |
 10% |
     | ...........     .......      .......      .......      .......          ...........
     +-----------+-----------+-----------+-----------+-----------+-----------+-----------+
     |           |   ppppp   |   ppppp   |   ppppp   |   ppppp   |   ppppp   |           |
     |    SUN    |    MON    |    TUE    |    WED    |    THU    |    FRI    |    SAT    |

                                 Each Column = 2 hours
                           (p = column includes prime time)

                     Legend                                 Prime Hours
                     ------                                 -----------
   "O" = Average               "." = Data Unavailable       0800-1700

                                                            Prime Days
                                                            ----------
                                                            MON,TUE,WED,THU,FRI

Notes:
The graph AVERAGE is at 45.9%, with a MINIMUM value of 34.8%, and a
MAXIMUM value of 57.7%.

Percentage of intervals with data for each node:
100.0% for GREEN         100.0% for BLUE        100.0% for AQUA
```

The following steps describe an approach to isolating disk contention using VAX SPM:

1 Initiate data collections during a time of typical system load using the following command:

```
$ PERFORMANCE COLLECT=TUNE/CLASS=(PROCESS,FILE_PRIMITIVES) -
 /INTERVAL=15/END="+1"/OUT=.INVDISK *
```

This command collects the default statistics (CPU, Disk, IO, Memory, XQP, Pagefault), standard process metrics, and FCP information for every 15 seconds during 1 hour on all nodes in a VAXcluster system.

2 Archive the data using the following command:

```
$ PERFORMANCE ARCHIVE=HISTORY/CREATE/INTERVAL=5 HISTORY.DAT *.INVDISK
```

This command archives CPU, memory IO, page fault, XQP, and disk data only. Process metric data cannot be archived into a history file and Process Metrics reports must be generated from individual log files.

3 Use a REPORT=HISTORY command of the following type to generate graphs for disks with a response time of 50 milliseconds or greater for 50% of the time:

```
$ PERFORMANCE REPORT=HISTORY/CLUSTER/NOTABULAR/GRAPH=D_RESPONSE:50 -
 /SELECT=GREATER_THAN:50/GENERAL HISTORY.DAT
```

Note that the /GENERAL qualifier specifies that each interval in the history file is reported.

4 If excessive disk activity is chronic, you can invoke the PERFORMANCE DISPLAY=FILES command as described in Chapter 9 to identify the most active files on the disk. If the activity of these files is the cause of high response times for the disk, you may move these files to other disks. It is important to continue evaluating the activity on these disks, however, before making any adjustments.

5 Create a Cluster Tabular report for those disks for which a graph was produced. Use a REPORT=HISTORY command of the following type:

```
$ PERFORMANCE REPORT=HISTORY/CLUSTER/NOGRAPH -
/TABULAR=(BYCLUSTER,BYNODE) -
/CLASS=(NODEFAULT_STATISTICS,DISK) -
/DISK=(disk1,disk2,disk3,...) -
/GENERAL HISTORY.DAT
```

A report similar to the one in Figure 21–6 is produced.

For the purpose of this example, disk $255$DUA1 is assumed to be a data disk that also accommodates the paging file for node GREEN. The CLUSTER Disk Statistics section of the report shows the rates for individual disks. Disk $255$DUA1 has a response time of 85 and a rate per second (Rate/s) of 38.

The BY NODE Disk Statistics section shows each node's contribution to individual disk rates. For example, for disk $255$DUA1, node GREEN's rate per second for is 20.0, Node BLUE's rate per second is 3.1 and node AQUA's rate per second is 6.0. Therefore, node GREEN is responsible for most of the activity on disk $255$DUA1.

6 Evaluate disk rates in the BY NODE Disk Statistics section to identify which node or nodes are responsible for the I/O rate. Examine the paging, swapping, and I/O rates to identify the types of activity that may be responsible for the high I/O rate.

Figure 21–6 Cluster By Node, By Cluster Disk Report

```
Report Date: 25-JAN-1989  17:24                    VAX SPM V3.3       Page    1
************             GENERAL                            ***********************
***********            Final   Statistics                   ***********************
Data Analyzed: from   27-MAR-1987 08:00:00.00   to  27-MAR-1987 17:15:00.00

+------------------- CLUSTER Disk Statistics ----------------------+
|                                                                  |
|                                              Resp                |
|                  Paging  Swping    Rate      Time   Queue  Space |
|                    %       %       (/s)      (ms)  Length  Used % |
|                  ------  ------  ------    ------  ------  ------ |
| o $255$DUA0      22.2     3.6      2.1       31     0.1    92.8  |
| o $255$DUA1      20.0    30.0     38.1       85     0.0    89.0  |
| o $255$DUA2       1.1     0.0      1.9       33     0.1    82.2  |
| o $255$DUA3       0.5     0.0      0.6       33     0.0    59.7  |
| o $255$DUA4       0.0     0.0      0.2       51     0.0    77.0  |
| o $255$DUA5       0.0     0.0      0.0       25     0.0    48.3  |
| o $255$DUA6       0.3     0.0      0.0       32     0.0    80.2  |
+--------------- (o ➤ Indicates incomplete data) ------------------+

+------------------------- BY NODE Disk Statistics ------------------------+
|                                                                          |
|                     Work                          Serv   Resp            |
|                     Avail  Paging Swping  Rate   Time   Time Queue  Space|
|                      %       %      %     (/s)   (ms)   (ms) Length Used %|
|                    ------ ------ ------ ------ ------ ------------ ------|
| $255$DUA0                                                                |
|   98% GREEN (8700)   1.5   17.1    1.6    0.6    27     28   0.0   92.8 |
|   97% BLUE  (8700)   3.6   27.6    4.8    1.3    27     30   0.0   92.8 |
|   98% AQUA  (8700)   1.1    5.8    0.9    0.3    45     46   0.0   92.8 |
| $255$DUA1                                                                |
|   98% GREEN (8700)   0.2   31.0   41.0   20.0    40     90   0.0   89.0 |
|   97% BLUE  (8700)   0.2    0.0    0.0    3.1    19     80   0.0   89.0 |
|   98% AQUA  (8700)   0.2    0.0    0.0    6.0    20     85   0.0   89.0 |
| $255$DUA2                                                                |
|   98% GREEN (8700)   1.1    0.5    0.0    0.3    39     39   0.0   82.2 |
|   97% BLUE  (8700)   4.4    1.2    0.0    1.5    29     31   0.0   82.2 |
|   98% AQUA  (8700)   0.3    0.0    0.0    0.0    68     68   0.0   82.3 |
| $255$DUA3                                                                |
|   98% GREEN (8700)   0.3    0.0    0.0    0.0    36     38   0.0   59.7 |
|   97% BLUE  (8700)   1.4    0.5    0.0    0.5    26     27   0.0   59.7 |
|   98% AQUA  (8700)   0.2    0.0    0.0    0.0    19     20   0.0   59.7 |
| $255$DUA4                                                                |
|   98% GREEN (8700)   0.4    0.0    0.0    0.1    62     62   0.0   77.0 |
|   97% BLUE  (8700)   0.2    0.0    0.0    0.1    28     28   0.0   77.0 |
|   98% AQUA  (8700)   0.3    0.0    0.0    0.1    67     67   0.0   77.0 |
| $255$DUA5                                                                |
|   98% GREEN (8700)   0.2    0.0    0.0    0.0    40     35   0.0   48.6 |
|   97% BLUE  (8700)   0.0    0.0    0.0    0.0    20     20   0.0   48.6 |
|   98% AQUA  (8700)   0.2    0.0    0.0    0.0    35     29   0.0   47.6 |
| $255$DUA6                                                                |
|   98% GREEN (8700)   0.2    0.0    0.0    0.0    17     18   0.0   80.2 |
|   97% BLUE  (8700)   0.1    0.3    0.0    0.0    24     25   0.0   80.2 |
|   98% AQUA  (8700)   0.2    0.0    0.0    0.0    21     22   0.0   80.2 |
+--------------------------------------------------------------------------+
```

Note that swapping reported by VAX SPM in disk statistics does not represent the number of processes swapped out. Instead, it represents the number of pages the swapper writes from the modified page list to the page file.

By examining the BY NODE Disk Statistics Section of Figure 21–6, one can learn the following facts about disk $255$DUA1:

- For node GREEN
  - Space used is 89.0%
  - Paging is 31%
  - Swapping is 41%
  - I/O rate is 20
- The high percentage of space used indicates that this is a heavily used disk. If a disk is not fragmented, a high activity rate and a full capacity may not indicate a problem. A disk that is already fragmented and is filled to its capacity, however, will become increasingly more fragmented.
- Nodes BLUE and AQUA do not show any paging and swapping. This indicates that they do not have page files on this disk. Node GREEN's paging and swapping rates indicate it has a page file on this disk.
- Since 70% of node GREEN's I/O rate is caused by paging and swapping, moving node GREEN's page file to another disk would lower the rate on this disk by 14 I/Os per second.

7 If high disk response time is due to paging and swapping as in the preceding example, you can move the page file to a less heavily used volume.

Identify the less heavily used volumes with a REPORT=HISTORY command of the type below. This command generates graphs for those disks whose response time was 30 milliseconds or less for 70% of the time.

```
$ PERFORMANCE REPORT=HISTORY/CLUSTER/NOTABULAR/GRAPH=D_RESPONSE:70 -
 /SELECT=LESS_THAN:30/GENERAL HISTORY.DAT
```

For cases in which the page file is the only file on the disk and it is responsible for a high rate, it may be necessary to split the page and swap files, and create secondary page and swap files on a less active disk.

8 On disks that contain images and data, paging can be caused by image activations as well as by paging activity. Therefore, it is necessary to evaluate image activations (described in Chapter 8) as a way of balancing the disk workload.

In addition, installing images that are frequently activated may also decrease activity. Use the DCL Image Level accounting utility to identify which images are frequently activated on a disk.

22 Using VAX SPM on Remote Nodes

Chapter 3 describes starting, stopping, and inquiring about collections from any node in a VAXcluster system. Chapter 12 describes invoking the Resource and Investigate video displays from any node in a VAXcluster system.

You can also invoke collections and the Investigate and Resource video displays on nodes that are not part of a VAXcluster system, or that are part of other VAXcluster systems. There are three requirements for invoking VAX SPM on a remote node:

- The remote node must have a standard or collector version of VAX SPM.
- The invoking node must have privileges for running VAX SPM.
- The remote node must have a proxy file, which lists the invoking node and user.

The invoking user must have the following privileges:

- To stop, start, or inquire about collections on remote nodes, either the SETPRV privilege or all the following privileges are required:

  ```
  ALTPRI    CMKRNL    DETACH    PRMMBX
  PSWAPM    SYSLCK    SYSNAM    SYSPRV
  TMPMBX    WORLD     NETMBX
  ```

- To invoke the Resource and Investigate video displays in real-time mode, the privileges CMKRNL, TMPMBX, and NETMBX are required.

A proxy file resides on the remote node and identifies users who can invoke collections and/or the Investigate or Resource video displays on that node.

A proxy file may be defined in one of two ways:

- It may be called SPM$PROXY.DAT and exist in the default directory of the SPM account.
- It may be a file pointed to by the logical SPM$PROXY.

The lines in a proxy file must be uppercase and have the following format:

```
NODENAME::USERNAME
```

The following lines in a proxy file specify that only users USERA and USERB from node GREEN can invoke VAX SPM on this node:

```
GREEN::USERA
GREEN::USERB
```

If a node name is specified without a user name or with an * for the user name, then all users from that node can invoke VAX SPM on the remote node.

In the example below, all suitably privileged users from node BLUE and node YELLOW can invoke VAX SPM on the remote node. For node AQUA, however, only user USERA can invoke VAX SPM.

```
BLUE::
YELLOW::*
AQUA::USERA
```

To begin collection on a remote node, specify the name of the remote node as a value for the node name parameter as in the following examples:

```
$ PERFORMANCE COLLECT=TUNE PURPLE
$ PERFORMANCE COLLECT=TUNE/INQUIRE PURPLE
$ PERFORMANCE COLLECT=TUNE/STOP PURPLE
```

The above examples start, inquire about, and stop a collection on node PURPLE. Node-specific initialization commands on the remote node affect commands in the same manner they would affect commands on a local cluster node.

To invoke video displays on a remote node, specify the name of the remote node for the node name parameter as in the following examples:

```
$ PERFORMANCE DISPLAY=INVESTIGATE PURPLE
$ PERFORMANCE DISPLAY=RESOURCE PURPLE
```

The above commands invoke the Investigate and Resource video displays on node PURPLE. Display initialization files on the remote node affect the commands in the same manner they would affect commands on a local cluster node.

If a user whose name is not defined in the proxy file attempts to invoke VAX SPM, the following security message is generated in SYS$MANAGER:OPERATOR.LOG, and generated on the console:

```
%%%%%%%%%%%  OPCOM   6-FEB-1989 10:27:07.73  %%%%%%%%%%%
Message from user SPM on GREEN

DECnet object SPM$REMOTE received unauthorized connect request.
 Request came from NODE = AQUA, USERNAME = USERX
```

A Collector Version of VAX SPM

This appendix describes the Collector version of VAX SPM. This appendix lets you:

- Gain an overview of the collector version of VAX SPM.
- Learn to use the SPM$MANAGER.COM command procedure with the collector version.

A.1 Overview

VAX SPM is provided in two versions: a standard version containing full functionality and a collector version limited to collecting performance information. You can install the standard version on nodes where you require the full range of VAX SPM functionality and the collector versions on nodes where you need to collect and archive performance data.

Table A–1 lists the collector version functions, the commands to invoke them, and the chapters in this guide that describe them.

Table A–1 Collector Version Functions

Function	Command	Chapter
Collect Performance Data	PERFORMANCE COLLECT=TUNE PERFORMANCE COLLECT=CAPACITY	Chapter 3
Collect System PC Data	PERFORMANCE COLLECT=SYSTEM_PC	Chapter 11
Display File Activity	PERFORMANCE DISPLAY=FILES	Chapter 9
Collect Disk Space Data	PERFORMANCE REPORT=DISK_SPACE	Chapter 10
Archive Performance Data into a History File	PERFORMANCE ARCHIVE=HISTORY	Chapter 15
Convert Version 2.0 History Files to Version 3.0 format	PERFORMANCE CONVERT=HISTORY	Appendix B
Monitor System Activity	Event Trace Facility (ETF)	Chapter 19
Automatically Collect Performance Data	SPM$MANAGER.COM	Chapter 3

While you are running the collector version, if you type a command that is not in Table A–1, an error message is displayed.

The SPM$MANAGER.COM command procedure collects and archives data, but it does not evaluate and report on data as it does in the standard version of VAX SPM.

A.2 Using SPM$MANAGER.COM with the Collector Version

A VAXcluster system can have all nodes running the collector version of VAX SPM, or it can have some nodes running the standard version and some nodes running the collector version.

If all nodes are running the collector version, you can set up the SPM$MANAGER.COM command procedure to automatically collect and archive performance data. Before you can analyze or report this data, however, the log files must be copied to a node running the standard version of VAX SPM. The SPM$MANAGER.COM command procedure automatically copies log files the day after they are collected when you define a location for the LOGFILE_DESTINATION symbol. Define the location as shown in the following example:

```
LOGFILE_DESTINATION == "NODE::DISK:[DIRECTORY]
```

In the following example, all log files are copied to subdirectory [SPM.DATA] on the disk named DISK$ on node GREEN.

```
LOGFILE_DESTINATION == "GREEN::DISK$:[SPM.DATA]"
```

If your VAXcluster system has one node that is running the standard version of VAX SPM, you can run the SPM$MANAGER.COM command procedure on this node to automatically collect, evaluate, and report on data for all nodes in the cluster.

Use the BATCH_QUEUE symbol to specify a batch queue on a node running the standard version of VAX SPM. If no queue name is specified, the queue name SYS$BATCH on the current node is assumed.

In the following example, the SPM$MANAGER.COM command procedure is submitted to a batch queue called GREEN_SYS$BATCH.

```
BATCH_QUEUE == "GREEN_SYS$BATCH"
```

To begin automatic collections on your VAXcluster system, set up the SPM$MANAGER.COM command procedure as described in Chapter 3. Define the LOGFILE_DESTINATION symbol to automatically copy log files from collector nodes to nodes running the standard version for analyzing and reporting. Define the BATCH_QUEUE symbol to ensure the SPM$MANAGER.COM command procedure runs on a node running the standard version of VAX SPM, and thus evaluates performance data for all nodes.

B Converting V2.X History Files

The internal format of a history file was changed with Version 3.0 of VAX SPM. Prior version history files can be converted to the new format using the PERFORMANCE CONVERT=HISTORY command. The Conversion utility provides support for adding system ID, node name, and allocation class information during the conversion procedure. It does not provide support for altering the archive interval or adding new classes of data.

Since the only interval supported for Version 2.x history files was 15 minutes, the interval for the converted history file must also be 15 minutes. To take advantage of the new archive interval and data selection capability, the Version 3.0 history file must be created and populated with data from the original log files.

B.1 Converting a History File

To invoke the Conversion utility, give a command of the form:

```
PERFORMANCE CONVERT=HISTORY/qualifiers... v3-hist [v2-hist,...]
```

where v3-hist is the name of the output Version 3 history file, and v2-hist is one or more Version 2.x history files that are to be converted. Wild card characters in the input file names are supported. The command format, parameters, and qualifiers are summarized in Table B–1.

Table B–1 PERFORMANCE CONVERT=HISTORY Command Syntax

Command Format

```
PERFORMANCE CONVERT=HISTORY v3-hist v2-hist, . . .
```

Command Qualifier	Default
/[NO]ALLOCATION_ CLASS(=class:name, . . .)	/NOALLOCATION_CLASS
/BEFORE[=time	/BEFORE=end of file
/CLASS=[(item, . . .)]	/CLASS=DEFAULT_STATISTICS
/[NO]CREATE([=EXTEND_SIZE:blks, INITIAL_SIZE:blks])	/NOCREATE
/[NO]LOG	/NOLOG
/NODE_NAME=name	/NODE_NAME=name in input file
/SINCE[=time]	/SINCE=beginning of file
/[NO]SYSTEM_ID([LOW=value, HIGH=value])	/NOSYSTEM_ID

B.2 Optional Parameters and Qualifiers

If /CREATE is specified, a Version 2.x history file need not be given. In this case, an empty Version 3.x history file is created. If /CREATE is omitted from the command line, the Version 3 history file must already exist. The keywords EXTEND_SIZE and INITIAL_SIZE given with /CREATE specify the initial size and file extend size for the Version 3 history file. By default, both of these values are 3000 blocks.

The /SINCE and /BEFORE qualifiers can be used to select records from the Version 2.x history files on a time basis. Absolute or delta times, or a combination of them may be specified.

The /NODE_NAME qualifier can be used to supply an SCS node name that is missing in the input Version 2.x history file, or to override an existing node name. The node name given will also be used as a prefix for any disk or device name (potentially served in the cluster) in the input history file that is lacking a node name prefix. If /NODE_NAME is omitted, the node name in the input history file is used. This qualifier can be specified globally, affecting all input Version 2.x history files, or locally on a per history file basis.

The /SYSTEM_ID qualifier can be used to supply an SCS system ID that is missing in the input Version 2.x history file. If /SYSTEM_ID is omitted, a value of 0 is assumed for both the low and high system IDs. This qualifier can be specified globally, affecting all input Version 2.x history files, or locally on a per history file basis.

The /ALLOCATION_CLASS qualifier associates an allocation class value with a system node name. All disk and device names that are prefixed with the node name given by this qualifier will have their prefixes changed to the associated allocation class value in the Version 3.x history file.

Example B–1 Converting a Version 2.x History File to Version 3.x Format

```
$ PERFORMANCE CONVERT=HISTORY/NODE_NAME=GREEN/SYSTEM_ID=(LOW=2544)-
  /ALLOCATION_CLASS=(7:GREEN) V30HIST.DAT V20HIST.DAT
```

The Version 2.x history file V20HIST.DAT is converted to V30HIST.DAT in Version 3.x format. Disks that have no node prefix in V20HIST.DAT are given the prefix 7.

B.3 Converting History Files from Several Nodes

To obtain cluster-wide reports from a history file, the history file must include data from more than one node in a VAXcluster. In cases where each node in a VAXcluster has its own V2.x history file, these files can be converted to a single multinode V3.x history file using PERFORMANCE CONVERT=HISTORY.

During this conversion, the /ALLOCATION_CLASS qualifier is used globally to map each specific node name to an allocation class. The /NODE_NAME and /SYSTEM_ID qualifiers are used locally on each V2.x history file to supply node name and system identification for each node.

Example B–2 Converting History Files from Several Nodes

```
$ PERFORMANCE CONVERT=HISTORY/ALLOCATION_CLASS=(7:GREEN,7:BLUE) -
  V30HIST.DAT -
  V20GREEN.DAT/NODE_NAME=GREEN/SYSTEM_ID=(LOW=2544),-
  V20BLUE.DAT/NODE_NAME=BLUE/SYSTEM_ID=(LOW=2545)
```

The Version 2.x history files V20GREEN.DAT for node GREEN and V20BLUE.DAT for node BLUE are converted to the multinode history file V30HIST.DAT in Version 3.x format. Disks that have no node prefix in V20GREEN.DAT and V20BLUE.DAT are given the prefix 7.

Index

Reader's Comments

Guide to VAX SPM
AA-G139B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	**Excellent**	**Good**	**Fair**	**Poor**
Accuracy (software works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

I would like to see more/less ______________________________

____

____

What I like best about this manual is ______________________________

____

____

What I like least about this manual is ______________________________

____

____

I found the following errors in this manual:

Page	Description
______	______________________________
______	______________________________
______	______________________________
______	______________________________
______	______________________________

Additional comments or suggestions to improve this manual:

____

____

____

____

I am using **Version** ________ of the software this manual describes.

Name/Title ______________________________ Dept. ____________
Company ______________________________ Date ____________
Mailing Address ______________________________
______________________________ Phone ____________

FOLD HERE AND TAPE. DO NOT STAPLE.

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Advanced Technology Products & Services
110 Spitbrook Road, ZK03-3/Z34
Nashua, NH 03062

ATTENTION: Release Engineering

FOLD HERE AND TAPE. DO NOT STAPLE.