INTRODUCTION

In the last few years, SAS Institute Inc. has established a worldwide reputation for its computer performance evaluation tools. There are a number of reasons for our success, including the SAS® System's ability to read data from complex accounting sources, provisions for time and date formats, and the capacity for statistical analyses and graphic presentations. It is important to us that you, our users, be aware of the array of computer performance evaluation tools available within the SAS System. The first section of this paper evaluates the interrelationship between components of the SAS System and suggests possible solutions to a variety of performance evaluation problems. The second section discusses SAS/CPE® software released in the spring of 1989 for VMS® computer performance evaluation.

MXG® SOFTWARE: THE FOUNDATION

The mainframe computer performance evaluation package developed by SAS Institute is comprised of two main components, Barry Merrill's MXG product and the Institute's CPE Starter Set. The MXG product creates a data base which the Starter Set then accesses to generate reports on computer performance and resources.

To develop MXG software, a sophisticated method of capturing accounting data, Barry Merrill needed an implementation language that made computer performance data uniquely accessible. It is not surprising that Merrill chose the SAS System; its high level language handles complex records and yet is easy to modify, and its data management facilities make it easy to retrieve and manipulate data. In addition to these features, the statistical functions in the SAS System make it possible for the user both to analyze a system and to control the results of the analysis, allowing for the production of customer specified reports with as much or as little information as wanted.

Using the SAS DATA step, Merrill's MXG software interprets SMF (System Manager Facility) records and RMF (Resource Monitor Facility) records into a SAS data library called the Performance Data Base (PDB). The PDB is made up of many data sets containing accounting information which can then be formulated into a variety of types of reports.

The flexibility of MXG software enables you to customize your Performance Data Base to a considerable degree. It also requires you to answer a few crucial questions once the PDB is built. For example, will your use of the PDB be confined to performance analysis only, or will you want to keep track of task analysis as well? Or will you be using the PDB for accounting purposes? Answers to these questions will determine the data sets you will need from the PDB.

You must also consider the timeliness of your reports. Daily reports deal with questions like, "How was the CPU yesterday?" or, "What was the response time on TSO so slow at 1:00 p.m.?". Weekly evaluation will cover issues such as the number of jobs or the number of transactions run per week. Most CPE analysts heavily summarize monthly data. It can be used to highlight problem areas occurring in a typical month of CPU usage. Yearly reports are useful for predicting CPU usage growth and determining if data center objectives were met.

Once you have identified service level objectives, the next step is to familiarize yourself with the PDB data sets which deal with your objectives. An important component of the MXG installation process is running the job which collects raw data into the SAS data sets which make up the Performance Data Base. This job, called BUILDPDB, creates data sets with information about CPU utilization, paging activity, workload activity, channel path activity, and device activity. The BUILDPDB data sets also provide information on individual jobs, steps, and lines printed, as well as a summary of RMF intervals.

On the VM operating system, MXG software is currently available for both VM account data and VM monitor data which is the only source of response time measurement to CMS users. VM session records are collected at logoff time and spooled to a VM account file. A VM utility then collects these records and presents them to the reader on a VM machine. The VM machine, in turn, collects and sends these records to an MVS data set where the MXG software processes the records and creates the SAS data sets listed below. A complete description of PDB data sets for MVS and VM operating systems is provided in Merrill's Expanded Guide to Computer Performance Evaluation Using the SAS System, which accompanies the MXG software package.

CPE STARTER SET FOR MVS AND VM ENVIRONMENTS

SAS Institute developed the CPE Starter Set in order to allow you to begin creating performance evaluation reports quickly without having to learn SAS code. Written using base SAS software along with SAS/AF®, SAS/FSP® and SAS/GRAPH® software, the CPE Starter Set is an interactive application which uses full-screen panels and menus. The Starter Set does not attempt to teach concepts of computer performance evaluation but is directed more toward the user who understands what indicators may be important for the system in question. By following the menu paths in the Starter Set, you can produce a variety of reports without having to learn SAS code. Using menus and fill-in-the-blank data entry screens, even nonprogrammers can quickly generate standard CPE reports. These reports fall into two broad categories:

1. Daily and weekly time-based reports for monitoring the change in a variable. These reports can be produced in two formats: bar charts or tables.

2. Period reports, which deal with a complete analysis period. The CPE Starter Set provides this information in the form of plots or tables.

An added benefit of the Starter Set is the option to tailor its existing programs to a particular installation's needs. It is expected that after a few sessions using the CPE Starter Set, you will begin to modify the application and to steal techniques from the application to integrate into your own systems.

The Starter Set offers a fast entry into tuning, resource tracking, and cost accounting. Sample programs are provided to help you get started in job costing or accounting systems. While the examples do not address billing, which differs greatly from installation to installation, three major parts of a complete billing application are available. These are:
1. Building and maintaining a rate table
2. Creating account code selection masks
3. Applying rates from the rate table to the Performance Data Base to create the Costed File.

The rate tables are viewed and updated through SAS/FSP panels. They are the key to the development of revenue scenarios.

The data source for the Starter Set is the Performance Data Base created by MXG. Some of the data sets used by the Starter Set for MVS consist of the following:

- RMFINTRV - Resource utilization and workload statistics
- JOBS - TSO, STC and JOB statistics
- STEPS - Step level statistics
- IPLS - One record for each IPL of the system
- TYPE72 - Performance group resources
- TYPE74 - Device activity.

Some data sets created by MXG for VM contain the following information:

- VMSESSN - One observation for each virtual machine session containing resources (CPU and I/O) consumed
- VMUSRD A - One observation for each user data record created
- V MDEVICE - One observation for each device attached to a virtual machine
- VMLOGON - One observation for each failed logon attempt
- VMIDisk - One observation for each failed disk attempt
- VMV CIA - One observation for each SNA session on behalf of a virtual machine
- ANALVM - VM accounting card analysis.

**SAMPLE SESSION**

We are now going to step through a sample session using the Starter Set for MVS. Our PDB contains performance data collected at SAS Institute in August of 1988.

The first slide represents the Master Menu panel. When you first come into the Starter Set, it is necessary to specify your PDB data set name. So we will select option 2 to set our session parameters.

As you can see, the panel lets you request that high-resolution graphics be used for all graphs produced by the Starter Set. These graphs can be stored in a separate graphics catalog and replayed at a later time for management reports. You may choose to direct your output to a specific graphics device or sequential data set. The GOPTIONS area at the bottom of the screen can be used to specify any output device supported by SAS/GRAPH software. You can also set SAS System options to be in effect for the duration of your Starter Set session. We also fill in system ID, terminal type, and the start and end dates for our analyses. The system ID and even the dates may be changed later in the session without returning to this screen. When our parameters are set, we return to the Master Menu. We are now ready to produce our first report.
We now select the type of data from the PDB to use in our report. Choosing option "3" will identify RMF data about resource utilization.

Once the variable has been chosen, we must now specify other variables to be used in our report. We can request the time span for the RMF intervals; we will use "DAILY." The statistic we want to compute is 'MEAN' or average of CPU busy time. An output data set can be specified but for our purposes, we will leave that field blank so that our output will be directed to the terminal.
Scrolling to the next screen gives us an area to fill in titles or footnotes. Ending from the screen will submit the job.

And this is the generated report.

This report shows how busy the CPU was for each hour on the 25th of August in 1988. As might be expected, the busiest hour was around noon; the time when the CPU was fairly inactive was about 6:00 a.m.
This positive slope shows that the variables change in the same way and at the same time (i.e., when percent CPU busy is greater, there are more active batch initiators).

Now let's look at a tabular report that is non time-based. Let's find out who was the biggest user of our CPU time. We'll look at the number of EXCP's and the TCB CPU time per account or job. From the Performance Analysis Menu, we selected option 5. Other Tabular Reports. This time, we will select JOBS data that contains information about each job that ran in the system.

We can request a tabular report about jobs in the system. We fill in the category variable as JOB (the jobname) along with the names of the variables we want to see in our report. We'll use 'EXCPTOTL', total EXCP's issued by the job. This number represents the number of blocks of data transferred regardless of the block size. We will also use 'CPUCTBTM', or the TCB CPU time in seconds. This number represents the CPU time attributable to the task or jobs running. Again, 'DIR' is available if we need help with the variable names. We're summing these variables because we want the total time per account/job. And we can scroll to fill in a new title.

This is the first page of our report.

Now we will look at another sample report from the Starter Set, this one using the "Top n Type reports" option. It is common in many companies to review the heavy CPU jobs or I/O burners. We'll use STEP level detail for this report.

Again, we will use 'EXCPTOTL' for the number of data blocks transferred. We will also use the 'TYPETASK' variable which indicates whether we have a TSO session, a job, or a started task.

This report reveals the top 10 CPU jobs in EXCP's.
We will now turn our attention to Cost Analysis. From the Master Menu, we select option "4. Cost Analysis." The first task will be to examine the rate table. From the Cost Analysis Menu, we select option "2. Edit the Table of Rates."

The rate table contains charges for the various resources. The table is a SAS data set with one observation per system (as defined by the system ID). The Starter Set allows charges for batch, TSO, and started task work. You can specify charges for CPU, I/O to different devices and printed or punched lines.

The PDB allows for three shifts. Each job run is assigned a shift designation according to your shift definitions. Typically, they are "p," for Prime Shift, "n," for Night Shift, and "w," for Weekend Shift. A complete set of rates for each shift can be entered by marking "x" in the appropriate field and scrolling right.

We need to tell the Starter Set where we want to save the account file. On the screen, we are asked to fill in an OS data set name into which the account file will be saved. This file may either be a new file or a file which already exists.

We'll select accounts beginning with "GER" and "ITA" to accumulate rates for our German and Italian ids.
Submitting this job will apply the costs in the rate table to the PDB data, and the “Saved Account File” will be created to contain the costed data.

This output represents the start of the “Saved Account File,” the basis of further reporting or billing activity. It can be quite large, depending on the amount of data in your PDB, so you may choose to create it with a batch job. You can do this by choosing option “7” from the Cost Analysis Menu to submit JCL to run a batch job.

Another useful feature in the Starter Set is the ability to run an interactive session separate from the menu choices available. Option “B” from the Master Menu allows you to drop into the SAS Display Manager System. Any code entered and submitted while in the Display Manager can be submitted to the SAS System and will be executed. This facility is useful for such functions as supporting nonstandard graphics devices where many GRAPH (SAS) software is to provide SAS programs that will process all important performance or accounting data and will support all performance related records written by any operating system or subsystem. To achieve this goal, MXG software offers methods for accumulating the various accounting records that it does not currently read directly. The accounting records written by DOS/VSE POWER provide similar to the step termination record written by the earliest version of SMF. MXG now processes this DOS/VSE POWER and CMS (CICS Monitor Facility) data. The MXG data sets which comprise the DOS/VSE accounting resource contain information on step resources, printing event resources, punch event resources, input card reader resources, and Remote Job Entry (SNA and non-SNA lines).

DB2 generates three different SMF record types, containing statistics accumulated during the DB2 system execution. Record type 100 contains statistics about overall DB2 system execution. Record type 101 has information about individual DB2 transactions. The MXG member, VMACDB2, reads these records. Type 102, containing detailed trace data, is processed by MXG member, VMAC102.

Merrill’s Guide offers a wide range of alternatives for accumulating data from the more common performance data sources as well as from some of the newer and less well-known data packages. At present, Merrill’s MXG software can read performance data from the following sources: MVS, MVS/XA, MVS/ESA, VM/370, VM/XA, DOS/VSE, and FACOM. If new data sources become available, or if you have measurement data not yet supported by MXG, contact your local SAS Institute office. Merrill may already plan to support this source. If not, and if you can provide documentation of the record structures and sample data, Merrill will frequently implement support for new sources.

CPE FOR MINICOMPUTERS

As recently as a few years ago, a system manager’s solution to performance problems on a VAX™ system might have been to just add another VAX. Nowadays, however, the investment required for the new, more powerful VAX machines is quite significant. By regularly monitoring your VAX machine, you can get maximum efficiency out of your current resources and plan ahead to ensure that future hardware acquisitions be included in your company’s budget. SAS Institute now offers a VAX computer performance evaluation product called SAS/CPE™ software. With SAS/CPE software, you can automate the collection and analysis of the massive amounts of data necessary to regularly evaluate your system’s performance and incorporate billing practices for use of your system.

As with other operating systems, the goal of computer performance evaluation on the minicomputer is to collect data over time and produce reports on that data at intervals. The data needs to be collected in a consistent format and kept available for analysis and reporting. With SAS/CPE software, you can collect data from any of four data-gathering facilities:

1. the VMS Monitor Utility (MONITOR)
2. the VAX Software Performance Monitor (SPM)
3. the VMS Accounting Utility (ACCOUNTING)
4. the SAS/CPE Disk quota facility (DISQUOTA).

MONITOR is a VMS utility that gathers system resources and performance data, as well as per-process and disk activity data. Reports are provided from which you can analyze your system’s consumption of CPU, memory, and I/O resources.
SPM, a VMS layered product, gathers performance information. SAS/CPE software lets you collect raw data through SPM and then process the data into SAS data sets. Reports on SPM data are not provided; you can write your own reports.

ACCOUNTING is a VMS utility that records system resource usage. ACCOUNTING data includes information on whether the usage was interactive, print, or batch. ACCOUNTING also provides per-user or per-image resource consumption statistics. You use its reports to analyze resource consumption and produce billing statements.

DISKQUOTA, provided by SAS Institute, specifies reports that summarize disk space usage by user, group, or disk. DISKQUOTA captures disk space usage data at specified intervals. In order for it to gather data, however, disk quotas in VMS must be enabled.

You can track or bill for disk space usage by incorporating DISKQUOTA data into the billing reports for ACCOUNTING data.

To collect and process the data, we use a utility called CPETOOL. It has the look and feel of any standard VMS utility. One command syntax collects the data, manages the data, and converts the data to SAS data sets. For long-term analysis, you can merge the data sets from multiple collections into groups. Variables that are not actually used in SAS/CPE software reporting programs are dropped out of these collections when they are merged into groups, making them more compact.

Once raw data has been processed into SAS data sets, you can use one of two interfaces to generate reports. Either interface can be called from CPETOOL with a single command. The interfaces are:

1. CPEMENU, a menu-driven SAS application that lets you choose reports interactively. It lets you create reports that are displayed on the terminal or stored in files, as well as replay reports that have been stored. You can have the reports created interactively or in a batch job.

2. Report Selection Facility (RSF), a command-driven subsection of CPETOOL. You can use it interactively or write command files that can be run in batch jobs. This latter capability makes it possible for you to automate the entire process of collecting data, processing data into SAS data sets, and generating reports.

SAMPLE SESSION

We are now going to step through a sample session using SAS/CPE software. We'll begin by collecting some MONITOR data. At the Command Line, we type "CPETOOL" then "create collection" and a name for the collection. In our case, we are calling the collection "mon_week" for weekly MONITOR data.

With the "create collection" command, we are telling the SAS System how to collect the data. We list separate options to specify the data source and the beginning and ending times for the collection, as well as how often we want to collect the data. For our example, the frequency for collection is 15 minutes. Also, note that the default on our system is to collect all classes of MONITOR data. The "start collection" command tells SAS/CPE software to run this MONITOR collection as a detached process. SAS/CPE software initiates the MONITOR utility, so MONITOR does not have to be running in the background.

Those of you familiar with VMS command syntax will notice how similar these command are to that standard format: command verb, parameter, and qualifier format following the "/".

When you issue the "create collection" command, SAS/CPE software automatically creates the directory structure and support files for collecting, managing and reporting on the data. All the raw data files, the SAS data sets and reports are stored in subdirectories that fall under the main directory which is noted by our collection name. In our example, the main directory is COLLECTION.MON_WEEK.

If we want to look at the characteristics of the collection, we type "show collection."
converted the raw data to SAS datasets (we are still in the process of collecting the data). "Disklist" is applicable only for Diskquota collections. "# Intervals" indicates the number of intervals to be collected and "# Remaining" indicates the number of intervals not yet collected. "File" provides the complete path name of the file being created. "Classes" specifies which classes of MONITOR data are being collected. The "Flags" category also indicates that the raw data is not yet in SAS data set form.

Once we have collected the data, the next step is to convert the data to SAS data sets to generate reports. We issue the command "process collection mon_week." This command submits a batch job to create the SAS data sets.

To view the characteristics of the collection now, again we type "show collection mon_week." Notice that we now have information under "data begin" and "data end." We are given the date time values that reflect the actual time that the data was collected. The date time value under "processed" is the date and time that we converted the data to SAS data sets.

To manage your collections, you can use additional commands to delete or rename collections or change characteristics. We also have the ability to combine one or more collections into a "group." For instance, we might want to collect MONITOR data once a week and then create a group for the month.

The ability to create groups provides two main advantages. First, it allows you to look at long-term trends in the data. Secondly, since SAS/CPE software drops variables that are not needed for long-range analysis, grouping allows you to reduce the data to a size more suitable for archiving. Should you wish to add reports which use variables that were dropped, it is simple to modify the code in the program which adds collections to groups using the SAS DROP and KEEP statements.

Now that we have collected the MONITOR data and read it into SAS data sets, we are ready to generate reports. We may use either the Report Selection Facility under CPETOOL or CPEMENU. These facilities allow you to produce written reports for MONITOR, ACCOUNTING, and DISKQUOTA data. The pre-written reports include summaries of resource usage, plots of resource usage over time, and billing reports.

Each interface can run interactively or in batch mode and will output to a line printer or graphics device. If you want to create high-resolution graphics, you will need SAS/GRAPH software.

We'll look at some examples of both interfaces. First, using the Report Selection Facility, we issue the command "report batch."

We specify the name of the collection, mon_week. The mode is graph so that we can take advantage of the high-resolution graphics in SAS/GRAPH software. We're using a stream-file format and a VT340 terminal. "Select" allows us to specify the type of report we want. We are requesting "fpl" for free page list report.

And this is the report produced.

Now let's look at CPEMENU. To access the menu facility, we type "report menu."

We specify the name of the collection, mon_week. The mode is graph so that we can take advantage of the high-resolution graphics in SAS/GRAPH software. We're using a stream-file format and a VT340 terminal. "Select" allows us to specify the type of report we want. We are requesting "fpl" for free page list report.

And this is the report produced.

Now let's look at CPEMENU. To access the menu facility, we type "report menu."
This is the Main Menu panel from which we can create reports, look at our system setup, or view previously created graphs. For our example, we will select option “1” to generate a report.

The next menu prompts us to enter the name of our collection or group.

Note that if we have forgotten the name, we can select the option to list all the available collections and groups. We know our collection name is mon_week so we enter it in the appropriate field.
And these are our graphs.

Now we'll take a look at ACCOUNTING data. Unlike MONITOR, SPM, and DISKQUOTA, the collection of ACCOUNTING data is started by the system manager from the DCL (Digital Command Language) prompt, so that the data is automatically collected by the operating system and stored in an accounting file.

SAS/CPE software also provides a sample billing application which contains examples of the files needed to produce a billing statement. There are two different ways to create the accounting file. One method is to issue the DCL command shown below.

```
$ SET ACCOUNTING/NEWFILE
```

When you issue this command, the current accounting file will be closed and a new one with a higher version number will be created.

The other method is to copy the data for a specific period into a different file and let the same accounting file continue collecting data. By default, the system accounting file is

```
"SYS$MANAGER:ACCOUNTING.DAT."
```

The command below (following the "$" sign) will extract the data for the month of August from the system accounting file and put it in a file called

```
"ADM:<RESOURCE>:ACCOUNTING.DAT."
```

The command below (following the "$" sign) will extract the data for the month of August from the system accounting file and put it in a file called

```
"ADM:<RESOURCE>:ACCOUNTING.DAT."
```

The command below (following the "$" sign) will extract the data for the month of August from the system accounting file and put it in a file called

```
"ADM:<RESOURCE>:ACCOUNTING.DAT."
```

The command below (following the "$" sign) will extract the data for the month of August from the system accounting file and put it in a file called

```
"ADM:<RESOURCE>:ACCOUNTING.DAT."
```

The command below (following the "$" sign) will extract the data for the month of August from the system accounting file and put it in a file called

```
"ADM:<RESOURCE>:ACCOUNTING.DAT."
```

We now create a new collection named "AUGUST" and use the "IMPORT" option to read in the accounting file created by the previous command. Note that when using the /IMPORT qualifier, we must also use the /VERSION qualifier to indicate the VMS version under which the accounting data were collected.

```
$ TYPE sasS,<:pe:<billimprates.dat
```

The rate table indicates the charge for various system resources. The table includes sample values that you will want to modify for your site. SAS/CPE software allows you to enter up to 20 different rate schedules. You can vary charges based on such factors as resource usage during prime versus nonprime hours, and batch versus interactive processing.
This is the report produced using the rate table we created:

```
REPORT:
/* rate table 1 */
RATE.NAM = "BASIC".
MP.MULT - .75 / multiplier for nonprime usage 
CPU.B - .088 / rate per batch CPU hour - prime 
CPU.P - .088 / rate per interactive CPU hour - prime 
CPU.B - .048 / rate per batch connect hour - prime 
CPU.P - .048 / rate per interactive connect hour - prime 
CPU.B - .160 / rate per batch connect hour - nonprime 
CPU.P - .160 / rate per interactive connect hour - nonprime 
MULTI.B - 1.00 / rate per interactive connect hour - nonprime 
MULTI.P - 1.00 / rate per interactive connect hour - nonprime 
FILE.P - 0 / rate per page - prime 
FILE.SP - 0 / rate per page - prime 
DISP.LP - .005 / rate per direct I/O - prime 
DISP.LP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O- nonprime 
FILE.P - 0 / rate per page - nonprime 
FILE.SP - 0 / rate per page - nonprime 
DISP.LP - .005 / rate per direct I/O - nonprime 
DISP.LP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O - nonprime 
FILE.P - 0 / rate per page - nonprime 
FILE.SP - 0 / rate per page - nonprime 
DISP.LP - .005 / rate per direct I/O - nonprime 
DISP.LP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O - nonprime 
FILE.P - 0 / rate per page - nonprime 
FILE.SP - 0 / rate per page - nonprime 
DISP.LP - .005 / rate per direct I/O - nonprime 
DISP.LP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O - nonprime 
FILE.P - 0 / rate per page - nonprime 
FILE.SP - 0 / rate per page - nonprime 
DISP.LP - .005 / rate per direct I/O - nonprime 
DISP.LP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O - nonprime 
DISP.NP - .005 / rate per direct I/O - nonprime
```

I hope with these examples, you now have a good feel for how SAS/CPE software can collect and present VMS resource and performance data. SAS/CPE software supports four different data sources: MONITOR, SP, ACCOUNTING, and DISKQUOTA. SAS/CPE software provides an easy-to-use interface for collecting and reporting on data. You only need to learn one command syntax to collect the data, and you have the choice of a command-driven or menu-driven interface. Finally, SAS/CPE software is flexible in that you can customize the product to meet the needs of your site.

**CONCLUSION**

SAS Institute is committed to providing computer performance evaluation tools to its users. The inherent features of SAS software make it the ideal basis for performance work in all SAS-supported operating system environments.

The data manipulation programs in Merrill's MXG product coupled with the user-friendliness of the CPE Starter Set for MVS and VMS provide our users with a basis for computer performance evaluation in the mainframe world. Work is in progress on the Version 6 Starter Set.

For minicomputers, SAS/CPE software affords a variety of reporting facilities and cost accounting procedures. The similarity in syntax of SAS/CPE software and the standard VMS command language make it easy to use and understand.

**REFERENCES:**


SAS, SAS/GRAPH, SAS/AF, and SAS/FSP are registered trademarks of SAS Institute Inc., Cary, NC, USA. SAS/CPE is a trademark of SAS Institute Inc.

MXG is a registered trademark of Merrill Consultants.