DETERMINING and PROFILING
MVS SYSTEM ACTIVITY with MXG® and SAS® SOFTWARE

Captain Paul J. Waldowski
United States Air Force

ABSTRACT
This paper describes a program written in the SAS system which determines and profiles daily resource utilization in terms of hourly service unit consumption in an MVS environment. The program utilizes Merrill's Expanded Guide (MXG) software as a frontend along with base SAS software, the SAS macro language, and SAS/GRAPH® software. Subsequently, each MVS installation is provided with a high-level view of actual system activity.

I. INTRODUCTION
In today's dynamic data processing environments, the most crucial endeavor facing capacity planners of a computer center running on IBM's Multiple Virtual Storage (MVS) operating system is that of providing decision makers with a high-level overview of how well the installation is operating. The main obstacles to accomplishing an endeavor of this magnitude include an overabundance of raw data that encumbers efficient measurement of the MVS system and the difficulty in producing trustworthy reports that can be used to draw quick and accurate conclusions about a computer's capacity. This issue is further complicated by the fact that uncaptured MVS overhead generated by various computing workloads must also be determined from the measured data collected in order to support critical reports about patterns of usage and overall consumption of existing resources.

The basic objective of this paper is to discuss contents and results of a SAS program which determines and profiles daily MVS system activity automatically. An automated approach is taken so computer professionals can devote their energies to analyzing and interpreting results, not gathering and calculating input factors.

II. PROGRAM DESCRIPTION
A. BACKGROUND
In 1985, the author entered graduate school at the University of Missouri-Rolla, Rolla, Missouri, to study computer science with an emphasis in operations research. It was during this time that the author not only completed a thesis entitled, "An Introduction to Implementing a Capacity Planning Effort in an MVS Environment" [2], but uncovered a number of new computer performance evaluation ideas concerning MVS environments as well. This study, in combination with his duties as an MVS performance systems analyst at various Department of Defense installations and his attendance at many vendor classes, has led the author to write a generic SAS program which determines and profiles daily MVS system activity automatically. An automated approach is taken so computer professionals can devote their energies to analyzing and interpreting results, not gathering and calculating input factors.
user objectives and forecast for tomorrow's data processing needs.

B. VERSIONS

Originally, the primary objective of this paper was to develop and discuss only a base SAS software version of this program. However, with the assistance of a colleague familiar with SAS/GRAPH and the availability of computer graphic facilities, a copy of the base SAS software version was easily converted to a SAS/GRAPH version. The two versions give all MVS installations an option, based on available facilities, of which one to use for performing capacity analysis.

Both versions are compatible and dependent upon Merrill's Expanded Guide (MXG) SAS software, produce data dependent SAS titles, detect and remove outliers, determine daily resource utilization in terms of hourly service unit consumption, and profile the relationships between measured and unaccounted CPU and I/O values. Additionally, they produce both vertical and horizontal charts (also called histograms) by employing the CHART procedure using base SAS software and the GCHART procedure with SAS/GRAPH software. They differ only in the quality of output each is capable of producing. The base SAS version output comes in the form of bulky cumbersome computer listings, whereas the SAS/GRAPH version output can be displayed from a graphics device and later printed in color. It is this type of presentation which can more easily convey to decision makers a high-level overview of how well the installation is operating. Nevertheless, it is the output generated by both versions which is supplying each MVS installation with actual patterns of usage and overall consumption of existing resources.

C. GETTING STARTED

The starting point for utilizing either version of this program begins with the creation of a MXG performance database commonly known as a TYPE7072 among MXG supporters. This SAS database is utilized as a front end for both versions of this program and can be created with any input SMF file using a single SAS statement. The statement, excluding the double quotes, is

```
%INCLUDE SOURCLIB(TYPE7072);
```

where SOURCLIB represents a purchased MXG partitioned dataset and TYPE7072 represents a member containing MXG software within the partitioned dataset. Alternatively, the performance database produced by the MXG supplied job called BUILDPDB could be used.

However, one of the main obstacles mentioned earlier was to reduce the amount of raw data that encumbers efficient measurement of the MVS system. Since this program was designed primarily for capacity planning purposes, selection of only SMF record types 70 and 72 was a natural alternative over building an entire performance database. This not only overcomes the obstacle of too much raw data but should prove to be useful to installation's who are interested in performing capacity planning studies concerning particular periods of time.

By default, both versions of this program assume that the performance database available as a front end contains at least the previous day's observations. This was done so an installation can automatically provide access to reports on how the installation was operating yesterday by simply spinning off a daily job using a renewable daily performance database built from the previous day's SMF recordings. For other requests, a user simply needs to supply the program with a date of interest. This can be done by invoking the macro called SMF_DATE with a parameter in the form of DDMMMYY, where DD is the day, MMM is the month, and YY is the year of interest. For example, the following SAS statement,  

```
%SMF_DATE(SMF_DATE='16AUG89');
```

would direct the program to generate reports reflecting the utilization of system resources on August 16, 1989. The only known drawback with this technique is that the front end performance database must contain observations with the date requested.

D. CALCULATIONS

Both versions of this program generate many nonstatistical calculations. However, only two of them are the key ingredients for displaying daily resource utilization in terms of hourly service unit consumption in an MVS environment. They are known as the hardware capacity of a computer system and the system capture ratio.

The hardware capacity of a computer system for our purposes is the number of work units a system could generate if the CPU and I/O were operating at 100% for an hour. This is not a new idea. It is part of Merrill's methodology [1:249-272] and can be implemented by using data from both RMF generated SMF record types 70 and 72. The calculation is purely algebraic in nature and the example provided below was developed in conscious imitation of the style adopted by Merrill [1:251-256].

CPU Model: IBM 3090 Model 200E
GIVEN: CPUCOEFF=10.0, IOCCOEFF=5.0, NRCPUS=2, SU_SEC=765.917

<table>
<thead>
<tr>
<th>MXG Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUCOEFF</td>
<td>TCB CPU Service Unit Coefficient</td>
</tr>
<tr>
<td>IOCCOEFF</td>
<td>I/O Service Unit Coefficient</td>
</tr>
<tr>
<td>NRCPUS</td>
<td>Number of CPUs</td>
</tr>
<tr>
<td>SU_SEC</td>
<td>Service Units Per Second of Task or SRB Execution Time</td>
</tr>
</tbody>
</table>

Two other constants are used in the following formula. They are 3600 and 1440000. The constant 3600 represents the maximum number of seconds one CPU would consume if it operated at 100% capacity for one hour. The constant 1440000 represents the maximum number of EXCPs (Execute Channel Programs) the I/O system would generate if operating at 100% capacity for one hour.

**FORMULA:**

One Hour Hardware Capacity =

\[
\text{Total CPU measured in seconds/Total CPU used in seconds} = \frac{\text{(CPUUNITS} + \text{SRBUNITS)}/\text{SU_SEC}/\text{CPUCOEFF}}{\text{CPU TCB CPU Service Units}} + \frac{\text{CPU TCB Service Units}}{\text{Number of CPUs}} + \frac{\text{CPU SRB Service Units}}{\text{Number of CPUs}} + \frac{\text{CPU SRM, and service calls}}{\text{SRM, and service calls}}
\]

Hence, after plugging in the given input values, an IBM 3090 Model 200E has a hardware capacity of 62,345,988 CPU and I/O service units per hour. In other words, this system is capable of processing approximately 62 million service units per hour. By using this measure, MVS archivists should be concerned when 85% of their resources are being utilized during their peak hours. This is often the first good sign that an out of capacity situation may be occurring soon.

On the other hand, the system capture ratio is an overall measure of what work processed in the system is "captured" by SMF record type 72 and what is missing. This concept is also not a new one. However, this idea has traditionally been applied to individual workloads. The approach taken in this paper is based on the assumption that unaccounted work presumably is going to the good of the system supporting functions such as paging, swapping, interrupt processing, I/O, SRM, and service calls. Having tested this approach on several systems with various workloads has convinced me that the approach is solid. Furthermore, this technique is taught in IBM’s class entitled “MVS/An Measurement and Tuning". Determining a capture ratio for a system is based on the following example.

**CPU Model:** IBM 3090 Model 200E

GIVEN: CPUUNITS = 1704548, NRCPUS = 2, SRBUNITS = 1004794, CPUCOEFF = 10.0, DURATM = 0.59:59.99, PCTCPUBY = 40.2, SU_SEC = 765.917

<table>
<thead>
<tr>
<th>MXG Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUCOEFF</td>
<td>TCB CPU Service Unit Coefficient</td>
</tr>
<tr>
<td>CPUUNITS</td>
<td>CPU TCB Service Units</td>
</tr>
<tr>
<td>DURATM</td>
<td>Duration of Interval (H:MM:SS)</td>
</tr>
<tr>
<td>NRCPUS</td>
<td>Number of CPUs</td>
</tr>
<tr>
<td>PCTCPUBY</td>
<td>All CPUs Percent Busy</td>
</tr>
<tr>
<td>SRBUNITS</td>
<td>CPU SRB Service Units</td>
</tr>
<tr>
<td>SU_SEC</td>
<td>Service Units Per Second of Task or SRB Execution Time</td>
</tr>
</tbody>
</table>

SAS Functions

HOUR - returns the hour from a SAS datetime

MINUTE - returns the minute from a SAS datetime

SECOND - returns the second from a SAS datetime

Three other constants are used in the following formula. They are 3600, 60, and 100.0. The constant 3600 represents the maximum number of seconds in one hour. The constant 60 represents the maximum number of seconds in one minute. The constant 100.0 is used to reduce the percentile expression, (PCTCPUBY/100.0), to a decimal term.

**FORMULAS:**

Total System Capture Ratio =

\[
\text{Total CPU measured in seconds}/\text{Total CPU used in seconds} = \frac{(\text{CPUUNITS} + \text{SRBUNITS})/\text{SU_SEC}/\text{CPUCOEFF}}{\text{CPU UNITS} + \text{CPU SRB Service Units}} + \frac{\text{CPU SRM, and service calls}}{\text{SRM, and service calls}}
\]

Hence, after plugging in the given input values, an IBM 3090 Model 200E has a total system capture ratio of 0.820465 for the hour observed. This value is then applied to raw type 72 records for the same hourly period in order to calculate the unmeasured service units the system generated to support the functions such as paging, swapping, interrupt processing, I/O, SRM, and service calls. The unmeasured service unit calculations are computed by dividing the measured service units from type 72 records by the total system capture ratio for each hour.

Once these calculations are determined, it turns out that the program has enough information available to profile daily resource utilization. However, before this is accomplished, the program scales all the computed calculations down to a 0-100 percentile range. This eliminates cluttering numeric values but more importantly allows for the presentation of the results in an understandable form to corporate decision makers. Furthermore, although all numeric values have
been scaled down, they are still available so a tactical computer performance evaluation (CPE) analyst can further investigate if capacity was lost due to system bottlenecks, inefficient use of resources, or over utilization of a resource.

E. OUTPUT VALIDATION

All outputs of this program should be analyzed, especially the management reports. Since these reports will inform corporate decision makers daily about how well the installation is operating, it is crucial that they correctly communicate accurate conclusions about a computer system's capacity. Although the program does remove outliers in the form of duplicate records, other error conditions are possible. Among the seven systems examined in this study, the biggest problem encountered was the incompleteness of RMF data for a particular system. This can be attributed for any number of reasons. The most common suspect being unexpected interruptions to RMF recording such as power failures. This places more responsibility upon the capacity planner to ensure that the input RMF data contains observations about every hour. Experience has shown that failing to quality control this aspect of the process will result in the form of unpredictable outcomes. My main point here is that my experience has been very positive if the RMF measurements are complete.

F. REPORTS

The most difficult part of any capacity planning analysis is to report how the computing resources are being used by users. In the past, users themselves have predicted their resource usage with some degree of success. However, their results are usually highly subjective and frankly, of dubious value, primarily because users are often people who may not have a computing background. These users of software often relate to programs as tools to help them accomplish their jobs without concern regarding what resources are being utilized. [2: p. 800]

Although it is true that users will continue to be an essential element to successful capacity planning, especially in providing estimates of anticipated future requirements, the key toward surviving in today's rapidly changing competitive business world revolves around the ways whereby an MVS installation can achieve meaningful forward estimates of the resources required, relative to the demand expected to be imposed by the workload. Obtaining such means will require measurement of actual workloads, knowledge of business plans and cycles, prediction of the impact of those plans, and timely communication of the results to management. Fortunately, the reports generated by this program contain what it takes to measure capacity that is accurate and in a manner which is relevant to management.

The reports produced by this program fall into three basic categories. These categories are known as CPU and I/O, CPU only, and I/O only. Each category is further broken down into subcategories known as measured and uncaptured service units, measured service units only, and uncaptured service units only. For those who have a MVS background, the CPU is further divided into SRB CPU time and TCB CPU time. Basically, SRB CPU time is time spent on behalf of a task by the system whereas TCB CPU time is time spent on the real work.

The first category, CPU and I/O, represent the core of the reports produced and will be the center of this discussion. All vertical chart reports in this category are labeled on the X axis with the hours in a day (0-23), where hour 0 is represented by measured and uncaptured service units consumed between the military time, 0000 through 0059, where the first two digits represent the hour in a day and the next two digits represent the minutes, hour 1 is represented by the military time 0100 through 0159 minutes, etc. The Y axis is labeled with percentile terms, 0 through 100 by 10, where 100 represents the maximum CPU and 1/O service units a particular system can generate in one hour. For example, the number 100, under the word SUM on the chart in Appendix A, represents the hardware capacity of 62,345,988 CPU and I/O service units that an IBM 3090 Model 200E could generate if it operated at 100% capacity for one hour. Horizontal chart report axes are labeled just the opposite. However, these reports have the added feature of the numeric value that each bar is representing on the far right of a chart (see Appendix B).

The other two categories, CPU only and I/O only, are merely subsets of the CPU and I/O category. On these charts, the number 100, under the word SUM, reflects either the maximum CPU service units a particular system can generate in one hour on the CPU only charts or the maximum I/O service units a particular system can generate in one hour on the I/O only charts. The other aspects of these reports are the same as those described above.

Finally, this program is just the beginning of capacity analysis. The next step is to write a program which handles a week's worth of data, probably the topic of my next paper. A week
provides opportunities to spot short-term trends and eliminates the inconsistency of monthly reporting due to the variable number of working days in each month. Furthermore, the work could be partitioned into prime and nonprime shifts, but the real analysis should be concerned only with the shift during which the configuration limits the ability to meet service. (1: pp. 267-268) For most MVS installations, these reports should supply enough valuable information to help set accurate user objectives and to increase the efficiency in forecasting for tomorrow's data processing needs.

III. CONCLUSIONS

The demand for a high-level overview of how well the installation is operating has long been a vital concern of many corporate decision makers. In dealing with this problem, this paper has provided trustworthy reports that can be used to draw quick and accurate conclusions about a computer's capacity. Furthermore, the program has determined the mysterious uncaptured MVS overhead with straightforward mathematical expressions as opposed to statistics in order to produce critical reports about patterns of usage and overall consumption of existing resources.

This program, as developed and evaluated, is intended to enhance the capabilities of MVS installations in identifying the point in time when the arrival rate of work will exceed the capacity of the available resources. It has achieved this by overcoming the major obstacles of not handling too much raw data and the resolution of unaccounted work. This was accomplished through selection of only RMF generated SMF record types 70 and 72 and the fact that the difference between hourly type 70 records and the sum of the same hourly type 72 records is the "uncaptured" service among all the activity within the system.

In closing, it may be stated with utmost confidence that the results produced by this program are open to no known legal objections, and that the advantages gained from this experience are sufficient to far outweigh any shortcomings it may contain.

ACKNOWLEDGMENTS

Many people have contributed to this paper in various ways, and their efforts and ideas are most sincerely appreciated.

I am especially indebted to my supervisor, Foster E. Miller, for providing me the time and support to develop and present this project.

Special recognition is given to colleague, Bradley Foxhall, for his technical assistance with SAS/GRAPH and most valuable suggestions.

I also wish to thank colleague, Michael J. Heffler, for testing and modifying this program to handle multiple systems simultaneously.

Finally, a very special acknowledgment is due to my whole family, especially my wife, Paula, for being a great source of support, encouragement, and for providing editorial assistance throughout all stages of writing this paper.

BIBLIOGRAPHY

APPENDIX A
SAS/GRAPH Version - Vertical Bar Chart Example
TOTAL MEASURED AND UNCAPTURED SERVICE UNITS BY SUBGROUP
DElivered on 10 JAN 90 BY THE
CPU MODEL - IBM 3080 MODEL 200E

APPENDIX B
SAS/GRAPH Version - Histogram Example
TOTAL MEASURED AND UNCAPTURED SERVICE UNITS BY SUBGROUP
DElivered by the
IBM 3080 MODEL 200E
10 JAN 90

297