Performance And Capacity Planning In A Mainframe Environment

Thomas G. Confrey, State of Connecticut
Ellie Rosenbaum, State of Connecticut

Abstract

SAS is a very efficient tool to analyze and report on performance and capacity elements in mainframe environments. This paper will concentrate on what data is readily available and some effective ways to present it to various levels of management. The techniques and rationales discussed here are not limited to mainframe environments only, but can be used as guidelines for reporting on other computers and components as well.

Background

SAS is a very efficient tool to analyze and report on performance and capacity elements in mainframe environments. Data can easily be extracted from System Management Facility (SMF) records and stored in a SAS database for later reporting. The storing and reporting can be accomplished with user written SAS programs or other third party code such as Merrill's MXG package.[1]

No matter what the method, once the data is available for use from a database or the SMF records themselves, decisions must be made on how to construct meaningful reports. This paper will concentrate on what data is readily available and some effective ways to present it to various levels of management. The techniques and rationale discussed here are not limited to mainframe environments alone, but can be used as guidelines for reporting on other computers and components as well.

In the large IBM mainframe environment almost all of the data relating to performance and capacity utilization can be found in the SMF records. In particular, the type 70-79 records produced by the Resource Monitoring Facility (RMF) contain an abundance of information. It is important to have a precise understanding of the data contained in the records. For instance, does one type of CPU utilization contain processor and operating system overhead? (Yes, the type 70 record contains both.) It is only then that one gains an appreciation for the types of reports and the level of detail that can be provided.

Questions

Before creating graphic presentations questions should be raised that will be useful in creating a professional product. Who are the reports for? Technical personnel who typically need detailed reports, or managers who may want more bottom line figures? What do the recipients intend to do with the information? Personnel responsible for performance will have different requirements than those interested in capacity planning. Also, what are the reports attempting to show? General trends? Specific transaction response times?

The amount and type of detail is also dependent on the time period that the report covers. It is not reasonable to expect the same amount of detail for a quarterly report as there would be for weekly or even monthly reports.

[1] A description of such a methodology can be found on page 447 in the SUGI 16 Proceedings (February 1991).

Lastly, how much data should be presented at once and in what fashion (e.g. graphic versus tabular)?

CPU Utilization

One of the most abused measurements in data processing is CPU utilization. Often it is depicted graphically as average daily utilization fitted with a linear regression line. There is no doubt this type of graph is useful in identifying utilization trends. However, there are some drawbacks inherent in this type of presentation. One is that the average, even when combined, or replaced, with maximum utilization is weak in that the peak times do not give a sense of how often the CPU was at, or near, the maximum. Another is when regression line graphs are presented, it is a natural inclination to extend the regression line out into the future to see when more capacity is needed. This line is not an accurate predictor. In this type of analysis there is not enough information contained in the graph to indicate how busy the machine was during the day other than its average and its maximum.

One method of extracting more meaningful information is to group the data from RMF intervals into percentiles.[2] Then by examining, say, the 90th percentile, one would know that 90 percent of the time the utilization was below a particular point. Figure 1 shows the daily percentiles of CPU utilization for all systems running on a processor. (This particular processor has three separate operating systems running on it through the use of logical partitioning. The CPU utilization, however, can be obtained from the type 70 record from any one of the systems.)

[2] RMF creates records based on an interval set by the installation. The data in an RMF record will cover one RMF interval. If the interval is 15 minutes then there would be four records for each hour with each record providing the average utilization for one 15 minute period.

The shaded areas represent the minimum, 90th percentile, and the maximum CPU utilization. If the graph is produced in color with the area between the 90th percentile and the maximum utilization shaded red it is much easier to read. Horizontal reference lines are drawn at 75 and 85 percent.

A variation of this appears in Figure 2 which contains data from one logical partition defined with three logical CPUs. Instead of CPU utilization percent, the Y-axis represents the percent of MIPS (Millions Of Instructions Per Second) consumed by this system. The number of MIPS that is available to this system is the number of CPUs allocated to it times the MIPS rating of one CPU. Rather than drawing a fixed reference line at, say, 52.5, the line is calculated by extracting the number of CPUs allocated to the system and multiplying it by this processor's CPU MIPS rating (e.g. 3*17.5). In this way the graph does not have to be adjusted manually if the configuration is changed.

Percentiles are also useful in quantifying resource usage over a long period of time. By reviewing the appropriate percentile for a month of CPU data (e.g. the 95th), it is easy to calculate the percent of the time the machine was over 95 percent utilized. It is then possible to determine the number of hours in the month that the processor (or system) was under stress.

A simple graph that can be very useful is shown in Figure 3. It displays the average CPU utilization by hour, Monday through Friday, along with the maximum utilization for any one RMF interval for that hour . It is a helpful summary of the amount of work being done on other shifts.

A very good graphic display that would be appropriate in a quarterly report is shown in

Figure 4. Here the average percent of CPU utilization by month covers periods where different processor models were in use. The three machines are denoted by the annotations 3090-200E, 3090-400E, and 3090-600J. The line traversing through the shaded area is utilization normalized to that of the 3090-600J and then forecasted from the first quarter of 1992 out to January 1994 with a 2.6% growth factor. The dotted grids in the background lend to the readability of the graph.

Utilization By Workgroup

At some point in time in the analysis of CPU utilization the curious will ask, "Who is doing all the work?" The type 72 record contains the amount of CPU (and SRB for I/O activity) service units consumed by each Performance Group (PG) in the system. If work is classified by PG it is then quite easy to summarize the amount of CPU resources consumed by converting the service units for each PG into CPU time. The type 72 records do not capture all CPU time and it is necessary to merge this data with data contained in the type 70 record.[3] The difference between the two is uncaptured CPU time. This difference can be reported separately, applied to each PG with either capture ratios or a percentage based on each PG utilization.

The pie chart in Figure 5 was constructed from information in the type 70 and 72 records. It breaks out the average utilization for the whole processor (three different logical partitions) by workload and calculates the average amount of unused capacity (18.2%).

Availability

Availability has become an important issue in many computer installations. There is an increasing amount of pressure to keep both hardware and software applications up 24 hours a day, 7 days a week. Hardware and software are currently designed with this goal in mind. If availability is an issue then it has to be reported and put into perspective. One perplexing problem involves quantifying down time (or the time a system or application is unavailable). Obviously, in many installations a CICS system that is down for one hour at 10:00 a.m. on a weekday has a much larger impact on the enterprise then it would if it were unavailable for one hour at 10:00 p.m.

The graph depicted in Figure 6 is one attempt to quantify and report such situations. Average CICS transaction rates for each hour are estimated from historical data and then used to estimate the amount of work lost due to an outage. In addition to graphing the actual data, there is a horizontal reference line indicating the objective (98.5% availability). There is also a line that represents the running average of availability percent. Annotation and a background grid add to the readability of the graph.

Response Times And Service Levels

Response times for transaction processing systems like CICS and TSO is an area where perspective is important. Instead of reporting just the average response time per transaction for a CICS subsystem, it is more meaningful to group the transactions into categories based on how long they take to complete. Examples of some of the categories might be all the transactions that finished in less than one second, or the number of transactions that took from one second to under two seconds to complete. A method of combining this information with service level objectives is shown in Figure 7. The objective in this case

[3] The 70 record contains the true amount of total CPU time and further breaks out overhead caused by logical partition management.

is that 98% of all transactions complete in less than five seconds.

The traditional metrics associated with TSO are first period transactions (i.e. trivial or short transactions), total transactions (both short and long), and the number of TSO users logged onto the system. TSO reporting is represented in Figure 8 with average and maximum response times for first period. The weakness in this graph is that it does not give an indication of how many transactions had good or bad response times. Even the median (50th percentile) would be a better gauge of response time. Since RMF does provide the standard deviation along with the mean, other methods are available to estimate the median.

Reporting

Once a database is in place it is not always easy for someone who is not intimately involved with the data to work with it. One way of addressing this problem is to use SAS/ASSIST which is a menu driven interface with the SAS system. It can be used to perform activities such as writing reports, analyzing data, creating graphics, as well as retrieving and storing data. This is accomplished by selecting items from different screens which then causes SAS/ASSIST to generate the necessary code, although it can be quite a cumbersome process.

Graphs can also be stored in a graphic catalog. A list of the graphs can be displayed by invoking a CLIST under TSO which calls SAS and issues the command for the PROC GREPLAY. Desired graphs can be displayed by selecting them from the list. It can be set up such that the user never has to enter a SAS command.

Summary

SAS is an effective tool to provide customized reports to different levels of your organization. A data base can be easily created from data readily available with unlimited options for graphic representation. In this cost conscious era it is more important than ever to provide accurate and meaningful figures. Taking a new slant on ever present issues may be all that is needed.

Acknowledgments

We would like to thank Pat Tierney and Susan Sullivan for their help. A special note of appreciation is extended to Art Bustria who couldn't be with us.

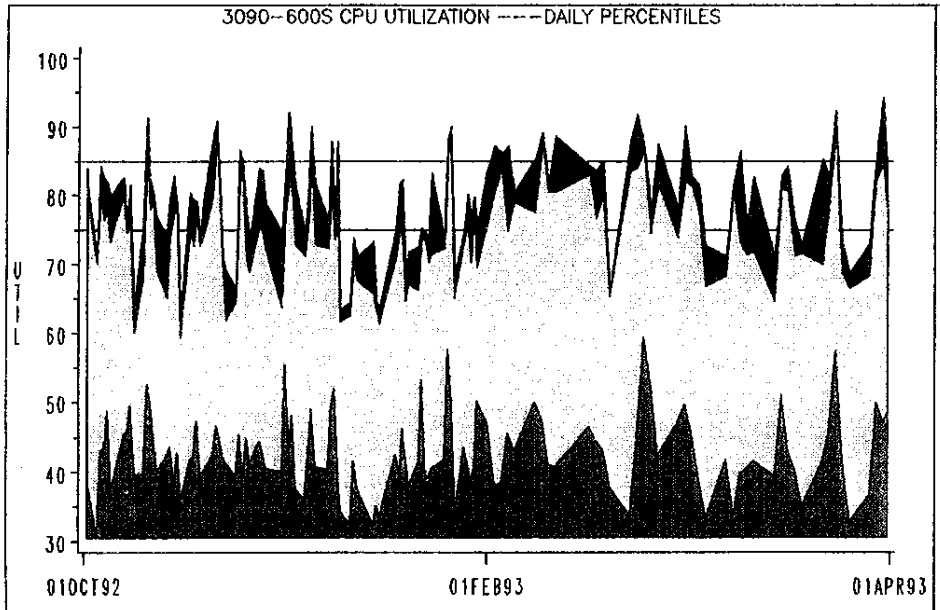


Figure 1.

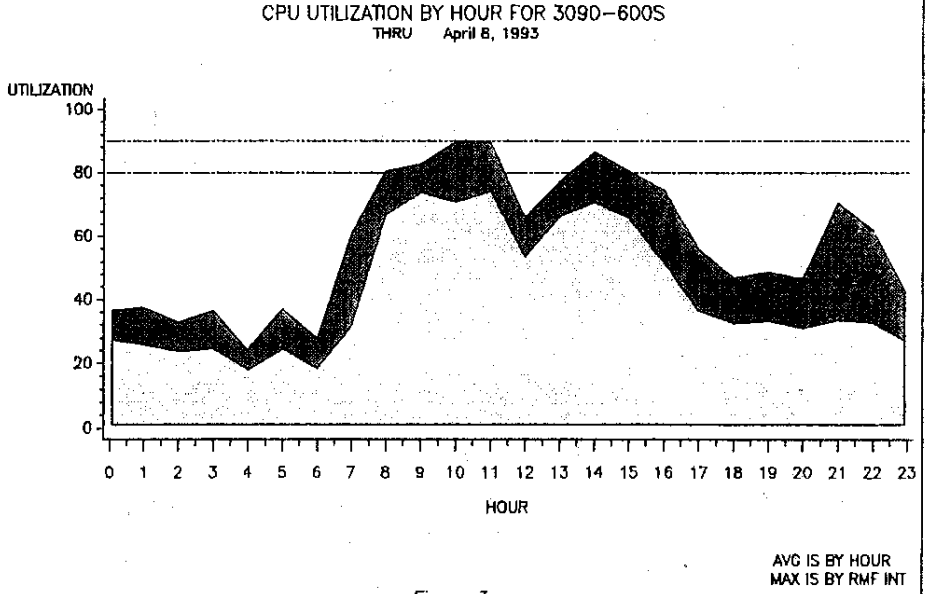


Figure 3.

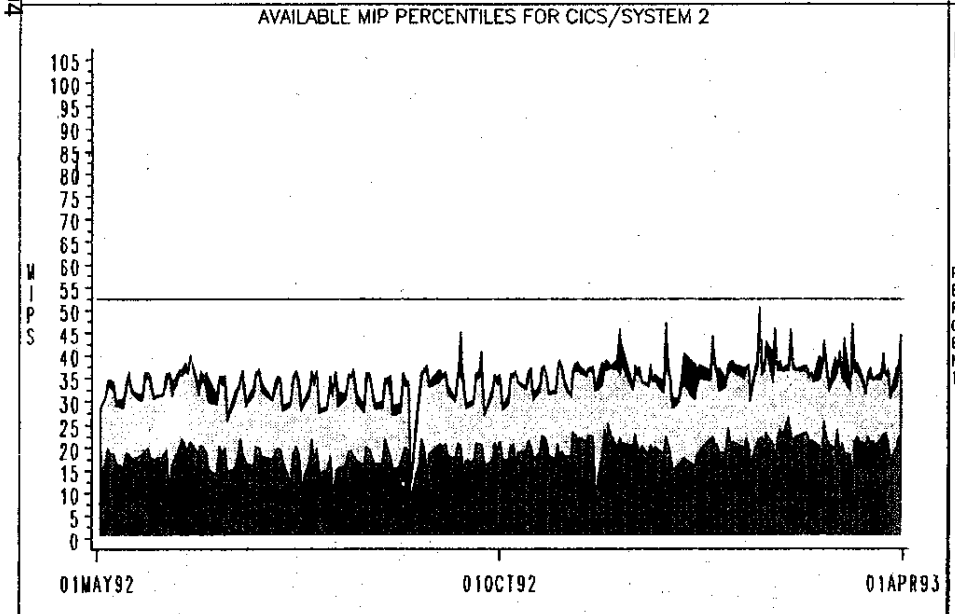


Figure 2.

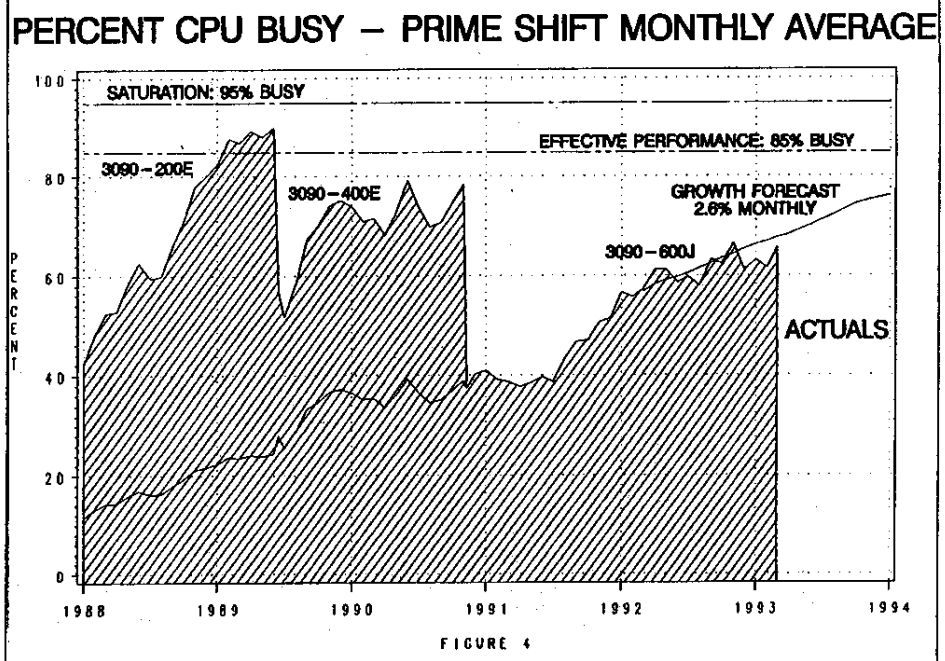


FIGURE 4

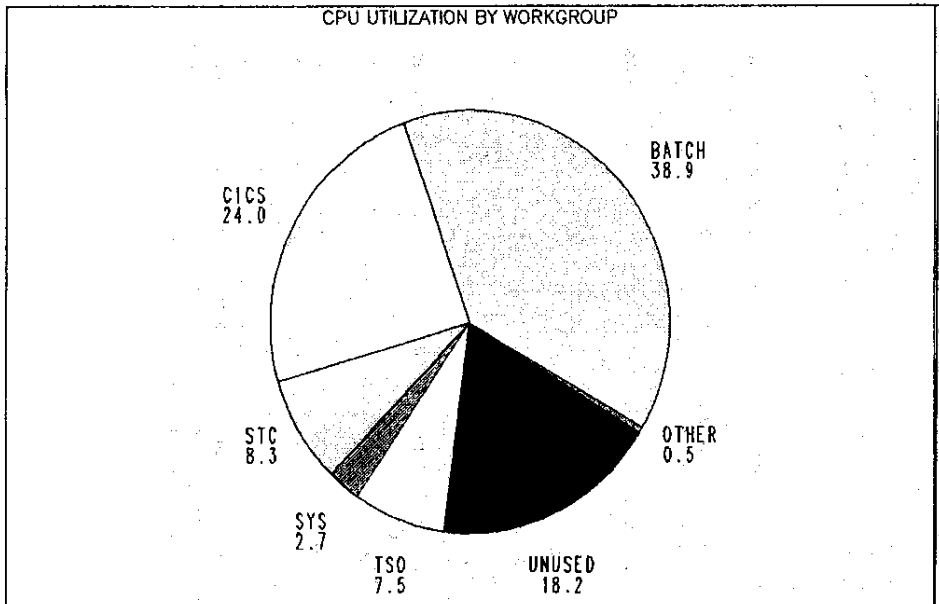
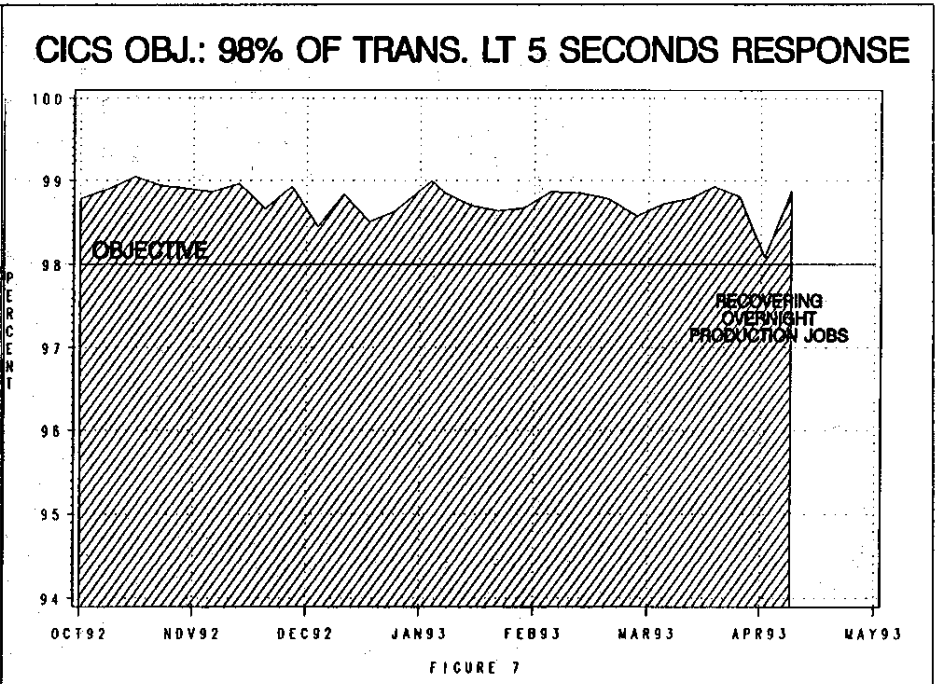
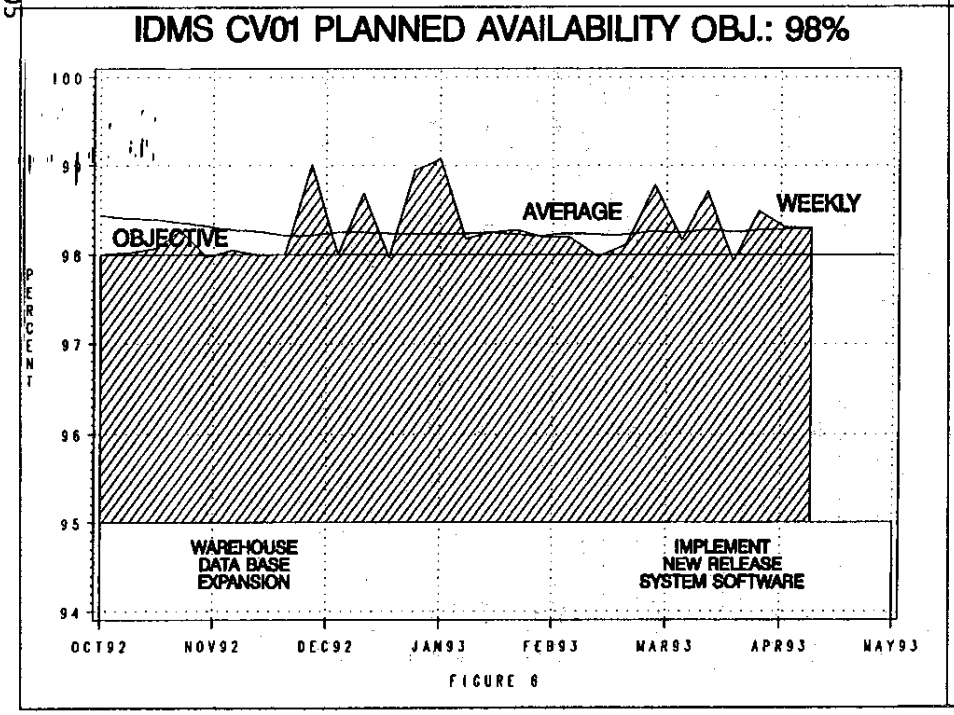


Figure 5.





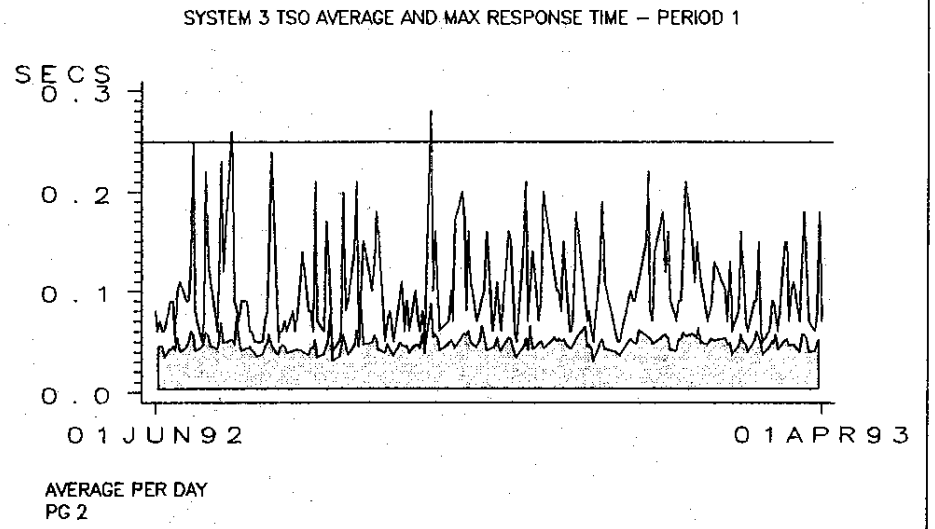


Figure 8.