PERFORMANCE MONITORING AND CAPACITY PLANNING ON MPE XL

Robert S. Apgood
Strategic Systems, Inc.
10502 11th Ave. NE
Seattle, Washington 98125
(206) 525-3309

I. INTRODUCTION

A vast number of excellent papers and seminars have been presented in recent years describing the steps to and complexities of Performance Monitoring and Capacity Planning in the classic HP 3000 environment. While most users acknowledge that the ideas presented in those papers and seminars would certainly be useful, not too many shops have actually taken steps to implement an on-going function of monitoring and analysis to evaluate current utilization levels and project trends of resource demands.

The intent of this paper is not to re-hash how it's done, what the numbers should look like, or what spiffy things our products do to make it easy for you to accomplish this task (even though that is true). Rather, we are going to look a common scenario and discuss why this needs to be done in the classic HP 3000 shop and, even more importantly (is that really a word?) why it must be done in the MPE XL environment.

This paper can be viewed as an "introduction" to the concepts and concerns of performing monitoring and analysis. It is the first in a series of papers to be presented by Strategic Systems, Inc. on the subject and will be followed in later conferences with progressively more technical and complex presentations.

II. TWENTY-TWENTY HINDSIGHT

Historically, in the MPE/V environment, most shops haven't really experienced the need for trend tracking. Why? The basic answer to this question is that they haven't really needed to. For the majority of its life, the HP 3000 hardware has been able to exceed the capabilities of the MPE operating system. Then, as the processing demands of most shops increased, the capabilities and performance of MPE kept pace with those demands and the user was lulled into a false sense of security that "trusty Old Clem" was actually able to handle the throughput demands all along. This feeling was further compounded by the fact that most shops experienced a need for greater connectivity (more terminals or

higher capacity discs) and upgraded to boxes that allowed this increase before (and in many cases well before) they ran out of other, more fixed in capability, resources (memory, CPU and channels).

Not heeding the indirect warnings coming from HP and users who had experienced similar phenomena in DP environments such as IBM shops, resource utilization tracking seemed unnecessary and no plans for implementing this tracking were made in the average HP shop. And then one day...

"HI! I'm your friendly neighborhood BRICK WALL!"

A look with your favorite monitoring tools shows you:

CPU UTILIZATION	=	99%
I/O UTILIZATION	=	87%
MEMORY UTILIZATION	=	98%
RESPONSE TIME	=	(Well, feel like taking an early lunch? You certainly have the time!)

So, now what? You call your HP sales rep and he tells you, "We can upgrade you to a Series 70!"

"Great!", you reply, "When can I take delivery?"

"September!", he cheerfully informs you, "I know it's a little ways away, but we've had quite a few orders for them recently."

Considering that today's May 13 (a Friday, wouldn't you know it), it seems that you're not the only ones who've had this little problem.

Sound familiar? Take solace in the fact that it's a frequent occurrence. But "what if" (poignant pause) you'd known for several months that you were going to run out of CPU sometime in May and you'd long since placed your upgrade order, and HP was installing it tomorrow...

What we're talking about here is nothing new. This same type of situation has been occurring in DP shops for years. In the HP world, in the IBM world, in the DEC world, in shops with every type of computer hardware ever made. The only difference is that very few of us in the HP world are truly prepared for it.

III. THE MINIMUM INFORMATION NEEDED

To effectively track resource utilization and response trends, a minimum set of information must be maintained over a period of

time. First, we must understand what the primary components of our systems are:

1) CPU
2) Disc I/O and Capacities
3) Memory

The effectiveness (or lack thereof) of the performance of each and all of these components are the contributors to the overall effectiveness of processing efficiency. The most apparent measurement of this effectiveness is the user's perceived response time. If the CPU is experiencing an abnormally high demand from the various processes executing, the perceived response time is lengthened. If an abnormally large number of disc I/O's are being required at the particular point in time, the requests are "queued" for servicing and the user must wait until the disc can service his/her particular request(s). A similar situation occurs when a process requires memory and none is currently available. When this happens, the Memory Manager is "awakened" and it is its duty to make the required space available to the process. If any (or commonly, a combination) of the above demands occurs, the user process is "stopped" until the request has been serviced.

So, it becomes rapidly apparent that tracking resource utilization over an extended period of time is required if we want to "see" how our processing demands are growing.

CPU:
Monitoring CPU utilization is probably the most critical and important of the major system resources. The reason for this is quite simple...it costs the most to replace. Luckily, except in rare circumstances (adding a new, major application, for example) does this utilization rate climb at a drastic rate. Therefore, planning for a CPU upgrade can easily be done in a time frame that allows for budgeting and ordering so as not to be caught off guard by the requirement. As stated before, however, this is the priciest of the major system components to enhance and, as such, close scrutiny to the performance of the other components should be made to determine that the bottleneck encountered or anticipated is, indeed, the processor.

DISC:
Disc is the second costliest resource to enhance. Unfortunately, it is frequently quite true that when the determination is made to add more disc to the system is required, the error factor is generally fairly low. Face it, if you need more disc space, you need more disc space. No surprises there. If you're looking for a rule of thumb, then if you've hit 85% disc space filled, you'd better call your sales rep. The assumption, of course, is that you've already purged @.@.GAMES, @.LASTYEAR.ACCTING and @.BACKUP.REFLECT.

DISC I/O:
Disc I/O is a frequently overlooked bottleneck. Unfortunately, Disc I/O bottlenecks have a nasty habit of disguising themselves as Memory Shortages. The last thing you want after waiting two weeks for your memory upgrade is to discover that your performance problem didn't go away. The common misconception is that since disc I/O rates are high, and Disc Caching is turned on, then Disc Caching must be the culprit and adding more memory will solve the problem (especially since that memory sales rep told you it would)! Now, don't take me wrong, more memory has its place, as I will discuss further, but right now we're looking at disc I/O. Entire papers and products have been written about disc caching, so I won't go into an extended discussion about it here. However, just as often as memory is the problem with disc I/O, so too is the caching configuration.

A second common problem with disc I/O is that the load placed on the discs is just too great for your current hardware configuration. This is particularly true with Series 70's. More than once has a system been configured with six 7933's all daisy-chained together nice and pretty and then hung on one GIC on one IMB. Sure, it's pretty...and pretty slow. As with Disc Caching, a multitude of papers has been presented on hardware and disc load balancing, so pursuing the details involved doesn't fall within the scope of this paper. The point being made is that before a decision to upgrade hardware is made, information must be gathered over an extended period of time and a careful analysis of the existing environment is mandatory.

MEMORY:
A shortage of memory will, without argument, cause a serious system performance degradation. If application programs are segmented poorly, or a large number of distinct processes are competing for the memory available, or the current disc caching configuration is causing a large number of write hits to dirty pages or a large number of cache domains are resident, then memory manager activity will definitely cause performance problems. However, most machines being sold today are generally intelligently configured with memory and another cause will frequently lead one to believe that a memory shortage exists. True, additional user load and/or new applications can create a need for additional memory. Unfortunately, not nearly so often as is suspected. Yes, more memory will hide a caching problem in many instances. However, a close look at and "twiddle" of caching fetch quantums will often "solve" a "memory shortage" problem. Again, more knowledgeable authors than I have addressed this syndrome in great detail and I refer you to their expertise.

IV. HOW THE "BIG BOYS" DO IT

Albeit new to the HP world, resource tracking and trend analysis is not new technology.

In one form or another, disc caching has been around for many years on other vendors hardware and fairly exacting methods for monitoring its effectiveness have been developed. The same holds true for memory utilization reporting, disc utilization reporting and response time analysis. What appears to many to be new technology, as some of you "Big Blue" expatriots well know, just isn't so.

So, what do we need to see?

RESPONSE TIMES vs. USERS

The best place to start is to determine what the "average" response time is based upon the number of users on the system. This type of trend analysis encapsulates the effectiveness of all the systems resources in the manner in which it affects the users. This historical information is valid on a daily, weekly, monthly, quarterly and annual basis.

As the "Response Times vs. Users" diagram indicates, it is possible to get a good feel for system efficiency just by tracking the average response time and mapping it into a predefined "acceptable" limit. The "acceptable" limit is a subjective response level defined by the system administrator. This is the level that the system administrator assigns as the maximum time in which a transaction must occur in order to accomplish the tasks of his/her environment.

Also provided by the system administrator is a "projected" response level, based upon past performance, that he/she anticipates the users will experience subject to the number of current users.

The "observed" response times for users are then tracked and mapped to this projection thereby providing the system administrator with a "snapshot" view of historical system performance. When a significant deviation is noted in the "observed" vs. "projected" slopes, the system administrator is quickly made aware of the fact and a more in-depth analysis can be made to ascertain the cause. Frequently, a deviation can be ignored if it still falls within the "projected" or "acceptable" limits. However, when this is not the case, subsequent reporting must be available to track the cause.

CPU UTILIZATION

Another, critical piece of information is an historical view of CPU utilization over an extended period of study. As with Response Times vs. Users, this report is useful for indicating peak CPU usage hours (daily), peak CPU usage days (weekly and

monthly) and, to a lesser degree, is useful on a quarterly and annual basis.

Tracking CPU utilization quickly indicates the peak demand periods for this critical resource and provides the system administrator with the information he/she needs to determine what steps can be taken to balance this demand.

As the "CPU Utilization" diagram indicates, tracking this information over the period of interest and computing an average CPU utilization demand gives invaluable insight into growing use of the processor. Tracking the "average" information on a quarterly or annual basis shows trends in increased CPU usage and allows for proactive steps in meeting a CPU saturation point as described in the above scenario.

DISC UTILIZATION

This is the most easily understood report and is self-describing. We note that disc utilization (disc space filled) is fairly static on most systems from day-to-day. In our example "Disc Space Utilization" diagram, we note that at the end of the month, a significant increase in space utilization occurs. In our example, we've added a new, large application and it demands a fair amount of all resources. Referring back to CPU Utilization and Response Times vs. Users, they support this observation. The rule-of-thumb here is, "If you try to put ten pounds of dirt in a five pound sack, it just doesn't fit. Period."

DISC I/O RATES

Whereas, Disc Space Utilization is an important subject of monitoring, Disc I/O Rates give a more informative description of the "pulse" of your disc I/O demands. Unfortunately, there are no "magic formulas" or rules that can be followed. This is due to the fact that virtually every machine has its own unique configuration. So, what is the ceiling on one Series 58 may well be the floor on another. This is particularly true in the Series 6x, 7x environment where you can have multiple IMB's, and a significant number of GIC's, which all can be processing disc I/O's effectively, simultaneously.

The diagram "Disc I/O's Per Second" shows that we've established an "acceptable" limit of 70 I/O's per second. This configuration is on a Series 48 with two 7933 drives each on their own GIC. At an average rate of 35 I/O's per second each, an "acceptable" rate of 70 I/O's per second total is within limits. If, on the other hand, You have a Series 70 with two IMB's, at a supported limit of two high-speed GIC's per IMB and a single Eagle drive on each of those GIC's, you could multiply the 70/sec rate by a factor of 2.5 and have a more-than-reasonable "acceptable" level.

So, you can see that this is certainly an environment specific mandated report. The concept, however, is applicable to any

business computing environment.

MEMORY UTILIZATION

Memory Utilization is, probably, the most misunderstood concept. Each vendor has it's own method of memory management and in some environments, the actual percentage of memory currently occupied is a valid measurement of memory requirements. This holds true on the classic HP 3000 for those systems that do not have disc caching nor AUTOALLOCATE turned on.

In our diagram "Memory Utilization", we note that memory utilization starts at approx. 91% and rapidly hits the 100% mark. This, in and of itself, doesn't necessarily indicate that we are experiencing a "memory pressure" situation. An initial glance at this report leads the experienced system manager to believe that AUTOALLOCATE is in effect and that further study is required. If, on the other hand, AUTOALLOCATE is not in effect, and a look at SHOWCACHE doesn't alarm us, then we probably need to assess our memory capacity and seriously consider adding more memory to the system, or modify our caching configuration to better utilize the memory currently available.

An excellent (excellent=inexpensive) method of assessing caching effectiveness is to use CDTMGR (contributed by Bryan Carroll, HP) found on the VEGAS swap tape. This little tool provides a real wealth of cache performance and lets you monitor the effectiveness of any cache configuration changes you make.

MEMORY MANAGER ACTIVITY

Probably the most accurate method of analyzing memory effectiveness is by monitoring Memory Manager activity. This means that you wish to observe how often the memory manager is required to "make room" for a particular structure (code, data, etc.). Although the measurement varies from vendor to vendor, the concept is the same regardless of the type of hardware you have. Whereas, on a FIREBLAST 6000, memory pressure is measured in nibble-faults-per-footpound; on a classic HP 3000, memory pressure is measured in cycles-per-second. Conceptually, the activity is the same; some indicator of memory management is used to measure how much time the Memory Manager is spending servicing the memory requirements of a particular process.

As a brief synopsis, on the classic HP 3000, when a request is made by a process for memory, the memory manager "cycles" through memory looking for available space to place the required structure in memory. The greater number of times (per second) that the Memory Manager needs to look through all of the memory available to find (or create) that space, the longer the process has to wait to continue.

As in the case of our example ("Memory Manager Cycles"), for the

majority of the month, Memory Manager "found" the space it needed with a minimum of cycling required to satisfy the request. However, we note, at the end of the month (after loading our new application) the number of times that MM had to cycle through memory increased drastically before a particular request could be completed. So, although Memory Utilization certainly didn't indicate that we were experiencing memory pressure, Memory Manager Activity certainly did!

V. HOW IT APPLIES TO MPE XL

Now that we have an historical perspective of the concepts of resource utilization tracking, along with an indication of how it applies to the classic HP 3000 environment, let's turn to the applicability of these concepts to the MPE XL environment.

In the MPE XL environment we have the following resources that contribute to the processing of a particular task:

1) CPU
2) DISC
3) MEMORY

So what's new? ...Nothing.

The only change in that processing is the manner in which certain tasks are performed and in the capabilities of the resources. CPU is faster. Disc I/O is faster. Memory is more abundant and addressability is greater. But the concepts are exactly the same!

RESPONSE TIMES vs. USERS
We still have online users who still have a perceived response time from the system. Therefore, we still need to track (and project) what that response time is based upon the number of current users.

CPU
We still have a CPU that is capable of performing a finite (albeit greater than the classic HP 3000) number of tasks in a given period of time. Therefore, we need to track the utilization of that processor over a period of time to determine what our historic needs have been, what our current needs are, and to project what our future needs will be. This becomes particularly important in the HPPA/MPE XL environment due to the cost and availability of upgrades. They're inevitable, so we need to anticipate when they must occur and plan and budget for them accordingly.

DISC UTILIZATION
The only surprise we may experience here is that we use up disc space faster than we did in the classic HP 3000 environment.

This is no surprise. In a RISC based architecture, the compilers generate more numerous instructions to accomplish a task than do their CISC counterparts. Additionally, we've grown into the HPPA environment, basically, due to the fact that the shear volume of our processing and data demands have increased. Additionally, with the advent of virtual mapping of data structures, among other items (such as the size of MPE XL, itself), our "virtual memory" requirements have increased.

DISC I/O RATES

Certainly, Disc I/O rates will increase due to the fact that we now enjoy the benefit of a much more efficient I/O facility. However, we can't ignore this aspect of resource monitoring. The concept of load balancing (both file locality and physical hardware configuration) doesn't change. If a channel or disc is busy, it's flat busy, and the hardware can only perform a single item of work at any single given instant.

MEMORY UTILIZATION

For the exact same reasons mentioned above, if there is memory available, it will be used. The only real change that we will note here is that the Disc Caching Facility (actually, the conceptual process) is now an integral part of the I/O and Memory facilities. To "change caching configuration" becomes a meaningless concept in that the function is, essentially, performed dynamically. The MPE XL experts out there, I'm sure, will take issue with the "imbedded caching" analogy. For their benefit, I confess that the technical implementation is far more sophisticated and pervasive than the analogy leads one to believe. Please bear in mind that the intent is to convey the concept and not to discuss the technical aspects of the implementation of that concept. So, to them, I extend my apologies and challenge them to make a presentation describing how, from a technical perspective, my analogy is meaningless.

MEMORY MANAGER ACTIVITY

As in any other vendor environment, measuring Memory Manager activity is still the best method of evaluating the effectiveness of the memory resource. If Memory Manager activity is high, something needs to change. A close scrutiny of the HPPA implementation clearly reveals that memory is a critical (!) resource to efficient throughput. I now start to support the memory proponents to a greater degree in the assumption that "More Memory Makes Bottlenecks Disappear." Although, as is true in any other environment, this is not an absolute, in the MPE XL environment it is an excellent **first** assumption. Series 9xx buyers take note: BUY LOTSA MEMORY!

VI. WHY YOU NEED TO START NOW

It has been the intent of this paper to make the reader aware of the necessity and benefits of performing an on-going function of resource monitoring and utilization trend analysis.

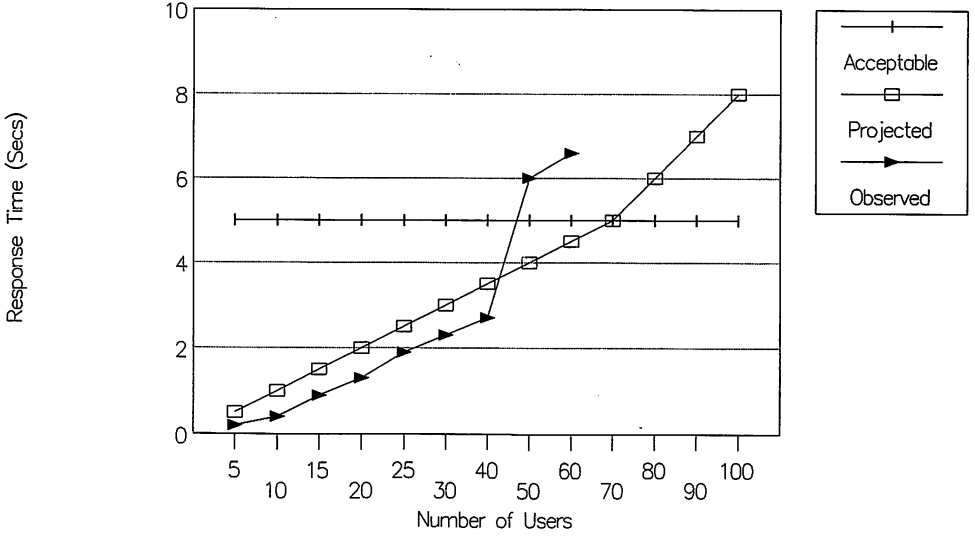
As has been pointed out, this function is applicable in any computing environment (yes, even on micros, to a certain extent). The idea is to implement a vehicle for informing the user of the use and abuse of available system resources and to have the information available that assists him/her in planning for future growth.

As our examples have shown, a fairly minimum set of information can provide enough data to the planning process to inform you in a timely manner as to when you can expect to increase the quantity or capability of a resource that is approaching a saturation point.

For those of you who are planning to upgrade to the HPPA environment, this can be of exceptional benefit. On the first hand, it may well show you that enhancing a particular resource may enable you to defer the upgrade for an extended period of time. Just as important, this information can give you concrete data for presentation to management to justify the upgrade for which you have been pushing. Either way, **knowing** what is happening on your system ensures that you're prepared to make whatever changes you will need to make in an informed, intelligent manner.

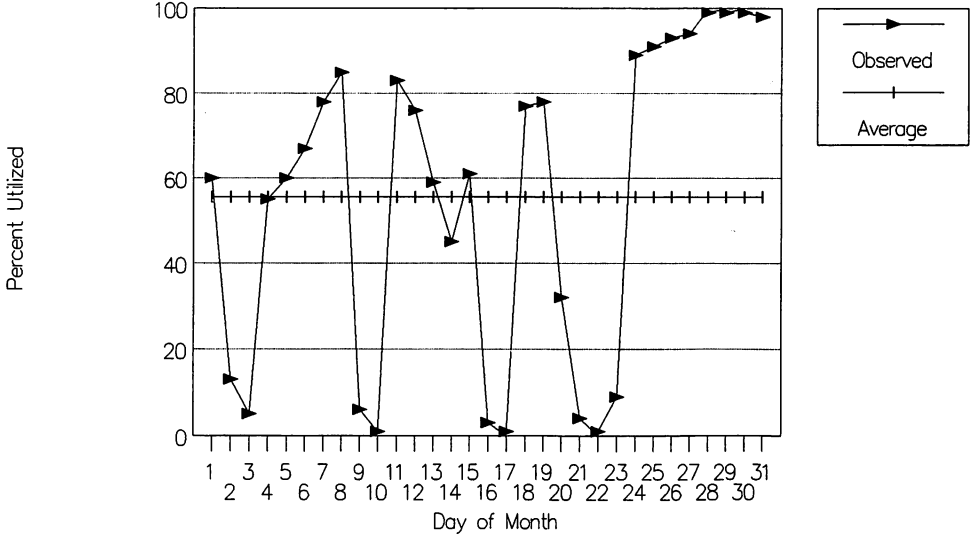
Response Times vs. Users

January '88



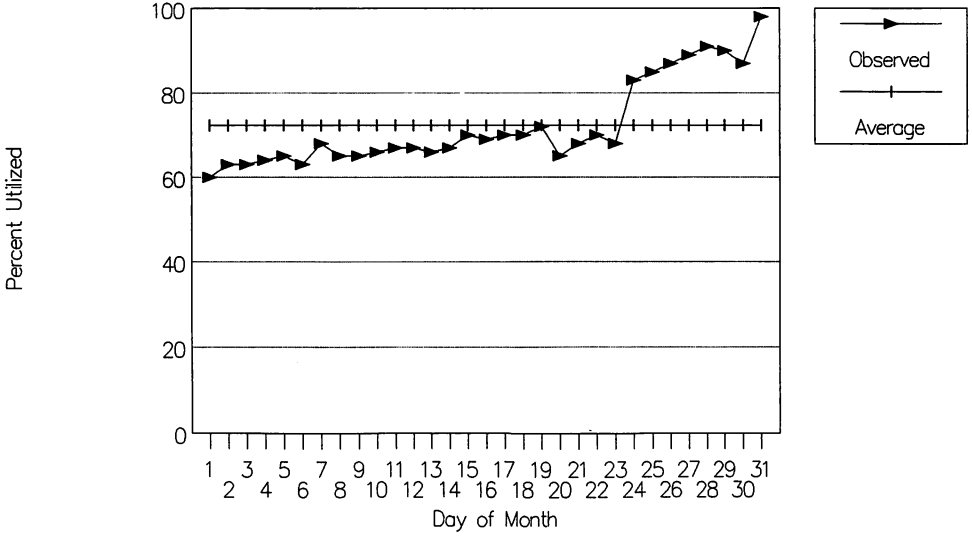
CPU Utilization

January '88



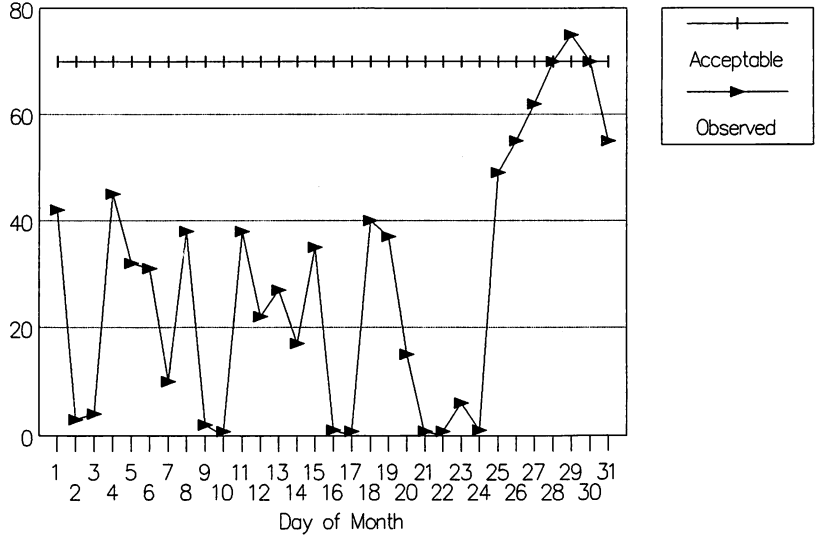
Disc Space Utilization

January '88



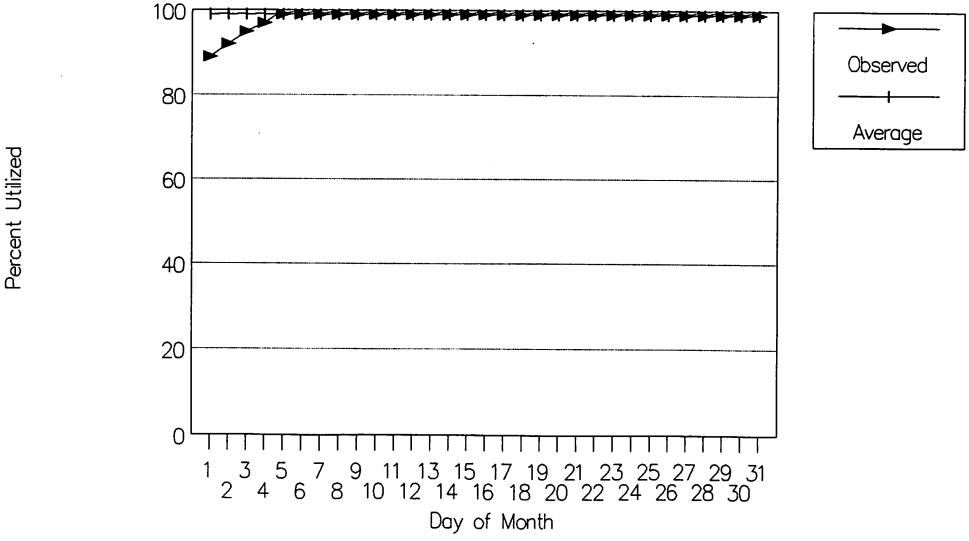
Disc I/O's Per Second

January '88



Memory Utilization

January '88



Memory Manager Cycles

January '88