IBM

Washington
Systems
Center

Technical
Bulletin

An MVS Tuning Perspective

By: R. M. Schardt
Washington Systems Center

GG22-9023-04
January 1981

Washington Systems Center
Gaithersburg, Maryland
Technical Bulletin

An MVS Tuning Perspective

R. M. Schardt

GG22-9023-04
January 1981

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to: IBM Washington Systems Center, Field Support, 18100 Frederick Pike, Gaithersburg, Md. 20760.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

SUMMARY OF AMENDMENTS

This is an update to GG22-9023-03. Major changes are indicated by a vertical line in the margin. These changes include:

- A discussion of the differences in the Device Activity Report produced for data collected on the IBM 3081 Processor Complex.
- Correction to the description of how RMF calculates Avg Q Lgn. Information in the previous version of this document was incorrect. Avg Q Lgn as reported by RMF includes request queued for all reasons not just Device Busy conditions.
- Terminology change of I/O Service Time to Response Time and Device Service Time to Service Time. These changes have been made to be consistent with other documentation.
- The section on I/O Contention has been reorganized and new information added.
- A discussion on IPS design has been added to the appendix.
- Foils have been updated to reflect the above changes.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENTS

A special thanks to Ron Clark and Brian Kasch who provided most of the MVS/SE information, to Siebo Friesenborg and Kurt Ziegler who provided much advice and council in the DASD tuning area, to Ram Vaswani for producing the SLR reports, and to Jerry Sly for his RMF expertise and the RMF data used in this paper. Also thanks to Paul Dorn who provided the discussion of the features of SAM-E that are included in the appendix. Their input was greatly appreciated.

THIS PAGE INTENTIONALLY LEFT BLANK

CONTENTS

LIST OF ILLUSTRATIONS

THIS PAGE INTENTIONALLY LEFT BLANK

INTRODUCTION

There are several requirements to be effective in tuning MVS. Among them are, knowledge of MVS operation, knowledge of the particular system to be tuned, and tuning experience. This paper addresses the latter requirement: experience. It does so by sharing with the reader the MVS tuning experiences of the DP Washington Systems Center staff.

Specifically this paper is intended to:

1. Suggest actions MVS installations can take to manage performance.

2. Provide an approach to tuning an MVS system.

In some respects this paper could be considered a primer, intended for the performance analyst who has been trained on MVS but may not have had much 'hands-on' experience. This information is also useful to managers responsible for system performance. The Chapter 'Performance Management' provides some suggestions as to the kind of activities that are needed to manage performance in a DP installation. The scope of this document is such that it does not discuss all the possible problems that could be encountered, but rather addresses those situations that tend to be common or reoccurring performance issues.

Included in this bulletin are a series of 'Rules of Thumb' (ROT) that can be used to help pinpoint the cause of actual or potential performance problems. The primary source for these ROTs is experience. Experience from studies and solving actual performance problems. Because they are based on experience, they are subject to change as experience grows. They are not absolute. These ROTs should be used along with knowledge of other system factors to determine the probable source of a performance problem.

Another aspect of these ROTs is that they are based on reasonable performance. In a given environment a ROT may be violated and yet performance is acceptable to that user. For example we know from experience that TSO response will suffer once channel utilization gets above 35% . Therefore the ROT states that channels servicing TSO volumes should be kept below 35% utilization. However, 'reasonable' response for a specific system may be less critical and a channel utilization of about 40% may be perfectly acceptable.

The whole point of this discussion is to point out the fact that the ROTs presented in this document are guidelines only. The real value of the ROTs is that they can assist in localizing the source of an actual or potential performance problem.

One last point. Although this document is directed primarily towards MVS, many of the ROTs may apply to any operating system. That's because many of the ROTs are related to I/O characteristics which are the same regardless of the software.

CATEGORIES OF PERFORMANCE PROBLEMS

Generally, the problems encountered in customer installations fall into three categories. They are:

1. Saturated CPU - CPU running at or near 100% and throughput is not acceptable.

2. Contention - Over committed real storage and/or DASD contention at the control unit or spindle level.

3. Performance Management

Of the three types of problems, causes of the first two are generally easier to find. That's because they are, for the most part, purely technical in nature and the experience level is such that most problems encountered today are not unique. However, the ease of implementing a solution can vary substantially. The fix could be as simple as changing a system parameter or as involved as a hardware upgrade such as more storage or additional DASD.

The most difficult of all situations to correct are those involving performance management. In many cases, these are environments where work has not been prioritized, no performance objectives exist, and the system isn't being monitored. These conditions must be corrected before performance problems can be addressed.

In the following pages each of the problem areas are discussed. The discussions will include symptoms of the problem along with some suggestions as to how the problem may be solved.

THIS PAGE INTENTIONALLY LEFT BLANK

In order to properly manage a DP installation certain standards should be established. In terms of system performance, several areas need to be addressed. First performance objectives must be set for the various kinds of work the system will process. In addition to providing users of the system with realistic performance expectations, the objectives establish a goal for system tuning.

Next, workloads need to be prioritized. This is necessary so that the system can determine which work to service first, especially during peak periods. Finally the system must be measured to establish if the performance objectives are being met.

Experience clearly indicates that the root cause of most performance crises is an inadequate performance management system. Unfortunately the need for a good performance management system is not always understood until it's too late. And trying to define an adequate system under the pressure of a crisis situation , seldom produces the desired results. The message is this; if an installation does not have documented performance objectives, has not prioritized its workloads, and is not monitoring system performance, it is more often than not, heading for a performance crisis.

The MVS Performance Notebook (GC28-0886) contains useful information on defining and measuring performance objectives. If you are not familiar with the Performance Notebook, review it before continuing here.

One activity mentioned in the Performance Notebook, that many installations appear to have a problem with, is work prioritization. The difficulty seems to be an inability or an unwillingness to decide what work is most important. The results of this indecision are particularly troublesome in those systems that are running at or near capacity.

There have been extreme cases where all work (TSO and batch) is run in a single domain or multiple domains with no discrimination between them. The potential problem with this kind of situation is that it tends to sneak up on you. So long as there are sufficient resources, the system may appear to perform properly. However, as the workload increases and the system reaches its capacity, performance problems begin to appear.

Generally speaking, when attempting to tune a system that has reached a resource limit, such as the CPU, it's an exercise in trade-offs. For example, TSO can be made more responsive, but it will probably effect batch thruput. Before these trade-off decisions can be made, the work must be prioritized.

PERFORMANCE MONITORING

Typically DP installations will do two kinds of measuring which are usually referred to as 'Long Term' and 'Short Term' measurements. Long Term measurements are those associated with performance monitoring functions. These measurements provide information about general system performance and workload trends. Long Term measurements are run continuously.

Short Term measurements, on the other hand, are generally associated with solving a specific performance problem or system tuning activity. Short Term measurements produce more specific and detailed information. For example, suppose the Long Term measurements showed TSO response time was increasing. Additional data may need to be collected to help identify the source of the problem. This latter set of measurements would be referred to as short term measurements.

RMF is flexible enough in its options that it can often be used both as a long and short term measurement tool. The following are suggestions for RMF parameters and the kind of data that should be tracked for both long and short term measurements.

RMF Parameters

CHAN
CPU
CYCLE (1000)
DEVICE (DASD)
ENQ (SUMMARY)
INTERVAL (60M)
PAGESP
PAGING
RECORD
NOREPORT
NOSTOP
NOTRACE
WKLD (PERIOD)
ARDJ/ASDJ (Major subsystems and components)

Recommended Data to Track

CPU Utilization
CHAN Utilization
DASD Device Utilization (Critical devices)
WORKLOADS (By Domain or Performance Group)
- Response Time
- Number of Transactions (Transaction Rate)

Paging Activity
- Page Faults
- DEMAND Paging
- SWAP
- VIO
- Page Data Set Utilization

Address Space Data
- CPU Utilization
- Storage
- I/O Activity
- Page Faults

Figure 1. RMF Long Term Measurement Recommendations

RMF Parameters

```
CHAN
CPU
CYCLE (250), (333) IF MVS/SE (see Note)
DEVICE (DASD)
INTERVAL (30M)
PAGESP
PAGING
RECORD
REPORT
NOSTOP
TRACE
WKLD (PERIOD)
ENQ (DETAIL)
ARDJ/ASDJ
```

Figure 2. RMF Short Term Measurement Recommendations

Note: Sixty samples (one sample per RMF cycle) of trace data are collected and displayed on a single line. The default cycle time is 250 milliseconds. For SU7 systems this default was fine since the SRM Resource Monitor routine was invoked every 30 seconds. However, in MVS/SE the RMR routine is invoked every 20 seconds. Therefore, a cycle of (333) is more appropriate since this results in a line of trace data every 20 seconds.

The major difference between the long and short term RMF parameters is the cycle time and interval. More frequent sampling and a shorter interval are required to capture more accurate data. A 30 minute interval for short term measurements works well for most situations. It's short enough to show shifts in workload and long enough to smooth out spikes that could be misleading.

Obviously the long term measurements with its less frequent sampling and 60 minute interval will put a smaller load on the system than the short term measurements. For example, sample measurements show that long term measurements driven by the RMF parameters described here impacted a 3032 CPU only about 1 to 2%. The impact of short term measurements will be greater and will depend on how many variables are being traced.

Performance monitoring and tuning are not one time activities but rather should be on-going functions. Most systems are not static. Workloads are constantly changing and new applications are being implemented. This kind of change can't help but effect system performance. A good monitoring system should help to anticipate and avoid problems rather than react to them.

To summarize, the major points relative to performance management are:

1. Performance objectives are essential. Properly stated they provide a clear definition of the service provided to the users and a basis for establishing system performance.

2. Workloads must be prioritized in order to establish the level of service to be provided to various users, especially during peak loads.

3. System performance must be constantly monitored. The data collected should be used:

   a. to determine if performance objectives are being met and, therefore, if tuning activity is required.

   b. to establish workload trends.

   c. as input to capacity planning activities.

THIS PAGE INTENTIONALLY LEFT BLANK

Many problems reported as poor MVS performance turn out to have nothing to do with MVS. Approximately 75% of the problems reported to the Washington Systems Center can be traced to some kind of I/O contention. Channel loading, control unit or device contention, data set placement, paging configurations, and shared DASD are the major culprits. As a general tuning rule, unless it is obvious that the problem is somewhere else, examine the I/O subsystem first.

RESPONSE TIME

The key to finding and fixing I/O related performance problems is response time. The length of time it takes to complete an I/O operation can have a dramatic effect on performance, particularly on-line and interactive subsystems such as TSO, IMS, CICS, etc. The following discussion addresses the various response time elements and the factors that can affect them.

For the purpose of this discussion, response time will be defined as shown in Figure 3.

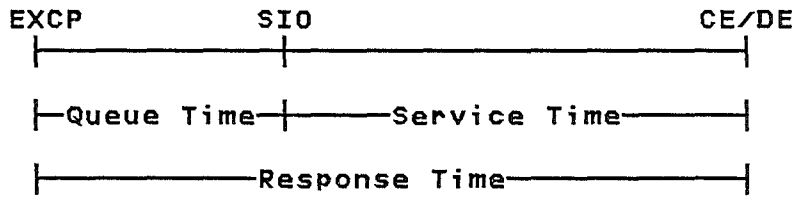


Figure 3. Response Time Definition

As shown in Figure 3, response time can be broken into component parts. Response time is the elapsed time from the execution of the EXCP Macro to the completion of the data transfer (Channel End/Device End Interrupt). It includes any queue time plus the actual I/O operation. Service time is the elapsed time from the successful SIO Instruction to the data transfer complete. It includes seek time, any rotation delays plus data transfer. Service time plus queue time equals response time. There are a number of factors that can affect response time. High channel, control unit, and/or device utilization can reduce the chances for a successful SIO. High channel utilization, control unit contention, and poor device seek patterns can further increase response time, once the I/O operation has

begun. Each of these factors will be examined in detail, but first some general response time Rules of Thumb (ROTs).

Queue Time

RMF reports Average Queue Length for a device. This value includes queuing caused by any busy condition. That is, queuing caused by device busy (UCB Busy bit on), channel busy, control unit busy and shared DASD/head of string busy.

Queue time is, therefore, Average Queue Length converted into time. To determine queue time, in seconds, divide Average Queue Length by Device Activity Rate (SIOs per second). Both of these values are reported by RMF.

As was stated, queue time consists of queuing for various reasons. Therefore, if queuing is excessive one must determine the specific cause. Queuing caused by control unit busy is reported by RMF in the field % CU DELAY. This value can be converted to time, in seconds, by dividing % CU DELAY by Device Activity Rate (SIOs per second).

Queuing caused by shared DASD and/or string switching contention is reported by RMF in the % RESV DELAY field. This value can also be converted to time, in seconds, by dividing % RESV DELAY by Device Activity Rate[1].

Queue time delays caused by channel busy are generally not significant. Therefore, they can usually be disregarded[2]. This means that any queue time not accounted for by % CU DELAY or % RESV DELAY can be assumed to be caused by device busy conditions (UCB BUSY bit on). Device busy conditions are an indication of the level of contention for a given device within a specific processor. High levels of queue time caused by device busy generally indicates two or more active data sets on a single device. The number of open data sets on a volume reported by RMF provides an indication of this situation.

The most significant queue time elements are % RESV DELAY and device busy (UCB BUSY bit on). Channel busy and control unit busy are significant but generally have a greater affect on service time than queue time. This subject is dealt with in greater detail in the sections on Channel Utilization and Control Unit Contention. But in general, the reason this is true

---

1 How RMF determines % CU DELAY and % RESV DELAY along with specific causes and cures for control unit and reserve delays are discussed in the subsequent sections of this document.

2 Refer to the section on Channel Utilization for more information.

is because if a request is queued for either channel busy or control unit busy, it will be delayed, on the average, one-half the data transfer time. Assuming a 6K blocksize on a 3350, data transfer is about 5 milliseconds. Therefore, if the request is delayed, this delay, on the average, would be 2.5 milliseconds. On the other hand, queuing delays for device busy can get as high a 50 or 60 milliseconds. Queuing delays caused by shared DASD can get into the 100 plus millisecond range.

To summarize, RMF reports Average Queue Length for each device. Average Queue Length can be converted to time, in seconds, by dividing by Device Activity Rate. The result of this calculation is called queue time. The individual components of queue time can also be determined from The RMF Device Activity Report. Queue time delays caused by control unit busy are reported in % CU DELAY. % CU DELAY can be converted to time, in seconds, by dividing by Device Activity Rate. Queue time delays caused by shared DASD or head of string are reported in % RESV DELAY. This also can be converted to time, in seconds by dividing by Device Activity Rate. Because queue time delays caused by channel busy are generally not significant, any queue time not accounted for by % CU DELAY or % RESV DELAY can be assumed to be caused by device busy conditions (UCB BUSY bit on). An example of these calculations can be found in the Problem Analysis Exercise Chapter.

Because queue time increases response time, it should be kept to a minimum. This is especially true for devices servicing interactive and online applications.

Some functions (e.g., ASM, TCAM) keep track of the active I/Os to the data sets they control. Therefore, assuming that these functions control all the data sets on a given device, there should be no queuing caused by device busy (UCB BUSY bit on). However, if significant average queue length is reported, look for high channel utilization, control unit contention or other active data sets on the volume.

Service Time

Service time is the elapsed time from the time a successful SIO (Condition Code 0) is issued until the Channel End Interrupt (or Channel End/Device End Interrupt) is received. Service time is affected by SEEK distance, control unit busy, channel busy, and, of course, blocksize.

Service time for 3330 and 3350 non-paging volumes should be somewhere between 25 and 40 milliseconds (paging volumes have a different set of ROTs and are discussed later). If these values are exceeded control unit contention, channel utilization and/or arm contention may be the cause. The version of RMF that supports MVS/SE provides service time in a field called Average

Service Time. For other versions of RMF service time can be calculated by dividing % Device Busy by Device Activity Rate.

3081 I/O Queuing

For SIOF[3] instructions, the 3081 Processor will accept the I/O request even though the path to the device is busy (i.e., channel busy, control unit busy, and shared DASD/head of string busy). If one of these busy conditions is encountered, the request will be queued in the hardware.

If after 10 milliseconds, a request queued in the processor has not been started, the hardware will attempt to restart the request. If unsuccessful, a deferred I/O interrupt is generated, a CSW is stored indicating the busy condition encountered, the request is dequeued from the hardware, and the request is queued by IOS.

The 10ms timeout applies only to control unit and device/head of string busy conditions. If the request is queued because of channel busy, it will remain queued within the hardware until the channel becomes free.

It is possible that within the 10 millisecond hardware queue time a request could encounter multiple busy conditions. For example, a request could encounter a control unit busy, and when the control unit became free and restart was attempted, encounter a head of string busy condition. In this case the CSW would indicate the last condition encountered (i.e., head of string busy).

The net effect of I/O Queuing is that the data RMF reports may be slightly different than what is currently observed on non-3081 systems. In those cases where no busy conditions exist, the I/O operation will not be queued in the hardware and there should be no difference in the RMF data.

When hardware queuing does take place but the request is started within 10 milliseconds, the service time for that I/O operation will include the hardware queue time. Because some requests will be queued in the hardware and some will not, the increase in average service time will probably be relatively small and therefore should not be a concern.

In those cases where the request was queued in the hardware but was not started within the 10 milliseconds, RMF is not aware of the busy condition for at least 10 milliseconds and therefore

---

[3] Starting with MVS SP1, the SIOF instruction will be used for all I/O operations except for SENSE, DASD RESERVE, and MODESET to tape.

the values reported for % CU DELAY and % RESV DELAY will be somewhat conservative. Because %CU DELAY and % RESV DELAY are included in queue time, it too will be somewhat conservative. However, these differences should not be significant. Realistically, one might expect to find % CU DELAY and % RESV DELAY to be conservative in a range of 2 to 4 milliseconds.

Whenever a CSW is stored at I/O interrupt time, the hardware stores information relative to the status of I/O hardware queuing in storage location 185 (decimal). Two fields are stored. They are:

1. Delay Code (Bits 0, 1). This field indicates what, if any, busy condition was encountered the first[4] time an attempt was made to initiate the I/O operation. Possible bit settings are:

   - 00 - No Busy Condition Encountered
   - 01 - Channel Busy
   - 10 - Control Unit Busy
   - 11 - Device Busy

2. Channel Queue Length (Bits 2, 3, 4). This field indicates the number of I/O requests queued in the hardware for this channel at the time of the Channel End Interrupt. It is a three bit field. All bits set to 1 indicate seven or more requests are queued.

Two new fields in the Device Activity Report report this information. % REQ CU DELAY is the percent of times the delay code indicated control unit busy. % REQ RESV DELAY is the percent of times the delay code indicated shared DASD/head of string busy. These values are useful in estimating the average delay time of the requests actually delayed by shared DASD/head of string and control unit busy.

Converting % CU DELAY and % RESV DELAY to time provides the average all I/O operations were delayed for shared DASD/head of string and/or control unit busy. In reality, only some I/O operations were actually delayed. The % REQ RESV DELAY and %

---

[4] As was mentioned earlier, it is possible that within the 10 millisecond hardware queue time a request could encounter multiple busy conditions. For example, a request could encounter a channel busy condition, and when the channel busy condition cleared and restart was attempted, encounter a head of string busy condition. In this case only the channel busy would be reported in the Delay Code.

REQ CU DELAY can provide an estimate of the average delay time of those requests actually delayed.

To determine the average delay for those requests actually delayed by control unit busy, divide the average delay time (% CU DELAY divided by Device Activity Rate) by % REQ CU DELAY. To determine the average delay for those requests actually delayed by shared DASD/ head of string busy. divide the average delay time (% RESV DELAY divided by Device Activity Rate) by % REQ RESV DELAY.

RMF will use the Channel Queue Length to calculate an average physical channel queue length and a distribution.

So far we have discussed the components of response time (queue time plus service time), how to break down queue time and convert the data presented by RMF into seconds, and some service time ROTs. Now let's examine some of the conditions that can cause response time to be excessive.

CHANNEL UTILIZATION

Channel utilization affects response time. As channel utilization increases, the response time for the devices attached to that channel increase. The impact of channel utilization is felt at several points during the execution of an I/O operation. The higher the channel utilization, the greater the probability the I/O request will be queued. These delays are relatively short and in most cases can be disregarded. That is because the probability of being delayed is equal to the channel utilization. And if the request is delayed, it will be delayed on the average of one-half the channel service time (i.e., data transfer time). For example, suppose the channel utilization is 35% and the average channel service time is 8 milliseconds. The chances of being queued are 3.5 in 10. If queued, the average queue time would be one-half of 8 or 4 milliseconds. Four milliseconds is generally not significant compared to total response time.

Channel utilization can have a major affect on service time. The higher the utilization, the greater the probability that attempts to reconnect after SEEK and SET SECTOR commands will be unsuccessful. If the reconnect is not successful after a SET SECTOR an entire revolution, 16.7 milliseconds for 3330/3350 devices, is required before the next attempt to access the data can be made. If unsuccessful on the second try, another revolution is required. Each time the reconnect is unsuccessful, a revolution is required. As the channel utilization increases, the number of unsuccessful reconnects increase, thus increasing the service time. For example, at 50% utilization about one extra revolution is taken, at 75% utilization three extra revolutions are taken. Because RMF reports average channel

utilization, it is likely at some point in time a channel showing 50% average utilization will hit 75% or better.

In general, channel utilization should not exceed 35%. Historically, this 35% rule comes from the fact that queuing theory indicates that at 35% path busy an I/O operation takes about twice as long to complete compared to a 0% path busy. However, this 35% rule is not really adequate because in reality, an acceptable level of channel utilization is a function of the subsystems that a given channel services. Online and interactive subsystems such as IMS and TSO require highly responsive I/O. Therefore, the utilization of the channels servicing their data should be lower. Generally speaking, TSO channel utilization should be kept below 35%. IMS channels should be kept below 30%.

I/O response for batch is generally less critical. Therefore higher channel utilizations are usually more tolerable. Channel utilizations of 40% or more are not necessarily bad.

Different levels of channel utilization suggests that the concept of channel balancing is not valid. Certainly if two or more channels are servicing the same subsystem, channel balancing is appropriate. However, one should not attempt to balance all channels where multiple subsystems are being serviced. In most environments, channel utilization should not be the same for response oriented subsystems as it is for batch. Each has its own requirement. The system should be tuned to those individual requirements.

CHANNEL CONFIGURATION

The point has been made that channel utilization should be a function of the subsystem it services. This suggests that in order to achieve these different response requirements, the I/O must be configured so that it can be tuned to the unique requirements of a given subsystem. It means that the I/O required for a given subsystem should be isolated to its own channel(s), control units, and devices.

The advantages of I/O isolation are significant. I/O performance at the subsystem level can be measured and tuned to specific requirements without affecting the performance of other subsystems. Where I/O isolation does not exist, tuning one subsystem often has an adverse effect on other subsystems. For example, take the case where both batch and IMS data are on the same channel. If you keep the channel utilization below 30% for IMS, full batch through put cannot be achieved. If you allow the channel utilization to satisfy the batch requirement, IMS response is likely to degrade. If you isolate, each can achieve its performance potential.

The amount of channel busy time required to READ a sequential data set can vary substantially depending upon the options selected for the data set. In general, sequential data sets should be defined as Fixed Block Standard (FBS). Where this is not possible, the DCB option of OPTCD=Z should be specified. (Refer MVS Data Management Services Guide, GC26-3875 for more information.) Another option is to install SAM-E (SU9). SAM-E forces a default of OPTCD=Z. The appendix of this document contains a description of the features and advantages of SAM-E.

If none of the above actions are taken, MVS uses a READ previous to access a given record. This means that in order to READ a given record, the previous record is READ, this is done to locate the selected record, and then the selected record is READ. In essence, two records are READ to get one. The result is twice the required channel busy time plus additional device busy. With OPTCD=Z, READ direct is used to access a given record. The channel program used by READ direct reads the selected record plus the Count field of the next record. The software then uses the information in the Count field to access the next record in sequence. This eliminates much of the unnecessary channel and device busy time.

Another aspect of channel configuring is to ensure that devices are properly sequenced on a given channel and the channel priorities are appropriate for the devices attached to them. This optimizes I/O performance by reducing the possibility of overruns. For DASD, an overrun causes additional rotations thus increasing the service time. A DASD overrun is generally caused by some kind of contention at the channel, or, in 303X Processors, the director. Correctly configured channels and directors minimize this contention.

A director on 303X Processors can generally achieve an aggregate block channel data rate of 6.7 megabytes/second with standard channels and 10 megabytes/second with the Data Streaming Feature attached with overruns causing only a negligible system effect. A high number of overruns at a data rate of less than 6.7 or 10 megabytes/second could indicate an improperly configured director. The '303X Channel Configuration Guidelines' (GG22-9020) should be reviewed for specifics on how to configure 303X channels. Although this document specifically addresses the 303X Processors, some of the information may be applicable to 158 and 168 systems.

DEVICE UTILIZATION

The traditional rules of thumb for device utilization have been similar to those for channels (35% rule). However, high device utilization, in and of itself, is not necessarily a problem. It can be a problem when it causes excessive queue lengths, particularly for response oriented subsystems like IMS and

TSO. Therefore, the measure of device 'goodness' or 'badness' is not utilization but queue time plus service time (response time). High device utilization can be one of the causes for excessive queue time. The higher the device utilization, the more likely the UCB BUSY bit will be on and, therefore, the request will be queued. High device utilization does not necessarily imply high service time. In fact, one frequently finds a DASD device with high utilization that has reasonable device service time.

CONTROL UNIT CONTENTION

DASD control unit contention, like channel utilization, affects both queue time and service time. It can cause busy conditions that delay the start of an I/O request, and delay reconnects after SEEK and SET SECTOR operations.

% CU DELAY and % RESV DELAY are determined by RMF via a sampling of bits set in the UCB. These bits are set based upon the condition code that results from the I/O instruction (SIO/SIOF). Figure 4 summarizes the possible conditions under which condition codes 0 and 1 are set. Also shown are the contents of the CSW, which bits in the UCB are set, and what conditions are reported by RMF.

SIO CONDITION CODE	CSW	UCBBSY	UCBCUB	RMF
0	0	1	0	% DEV BUSY
1	BUSY & SM	0	1	% CU DELAY
1	BUSY & NO SM	1	1	% RESV DELAY

Figure 4. SIO Condition Codes

There are essentially two conditions that can cause a condition code 1 on a SIO Instruction. Control unit busy which is indicated by the Busy and Status Modifier bits in the CSW and device or 'head of string' busy which is indicated by the Busy bit and no Status Modifier in the CSW. Let's examine control unit busy conditions first.

The control unit is busy during data transfer time. While data is being transferred the device, control unit and channel are essentially tied together for the duration of the data transfer. With only one channel path to the control unit, the probability of encountering a control unit busy is small unless there is a high level of FORMAT WRITE activity to 3330 Model I drives. The Model 1 drive does not release from the control unit while erasing the remainder of the track. Therefore, the

control unit and device are busy during this time but the channel is not.

There is also, with any RPS device, a short period of time between the reconnect resulting from a SET SECTOR and the start of the data transfer when the control unit is busy but the channel is not.

When the control unit is found busy, the software sets the Control Unit Busy bit on in the UCB.

As was discussed earlier, one can determine the average amount of time an I/O request is queued due to control unit busy from RMF data. Like channel utilization, the time a request is queued because of control unit busy is generally much less significant than the revolutions lost because a device could not reconnect due to control unit busy. And like channel utilization, these delays due to lost revolutions show up in service time. However, if the queue time due to control unit busy (% CU DELAY) is excessive, one way to get some insight into the problem is to calculate the response time, including all the elements of queue time, for each device attached to the given control unit. This analysis will probably show one of two conditions: too many high activity devices on a single control unit or one dominant volume. The latter condition is relatively easy to spot because a volume that monopolizes a control unit generally has a higher SIO rate with low response time while the other devices on that control unit have lower SIO rates and relatively high response time. What needs to be done in this situation is determine why the device is dominating the control unit. Perhaps if multiple data sets are involved some could be moved to devices on other less active control units.

Generally, RMF provides enough data to determine the cause of a control unit contention problem. However, there are situations where more detailed information is required. In these instances a GTF Trace and reduction may be required. The section on GTF Data Analysis contains some suggestions on how this might be accomplished.

SHARED DASD/HEAD OF STRING CONTENTION

% RESV DELAY can indicate shared DASD and/or head of string busy. It means that a SIO was issued and the channel and control unit were available but the device was busy or, when string switching is installed, the 'head of string' was busy.

Let's examine the situation without string switching first. Refer to Figure 5 on page 21.

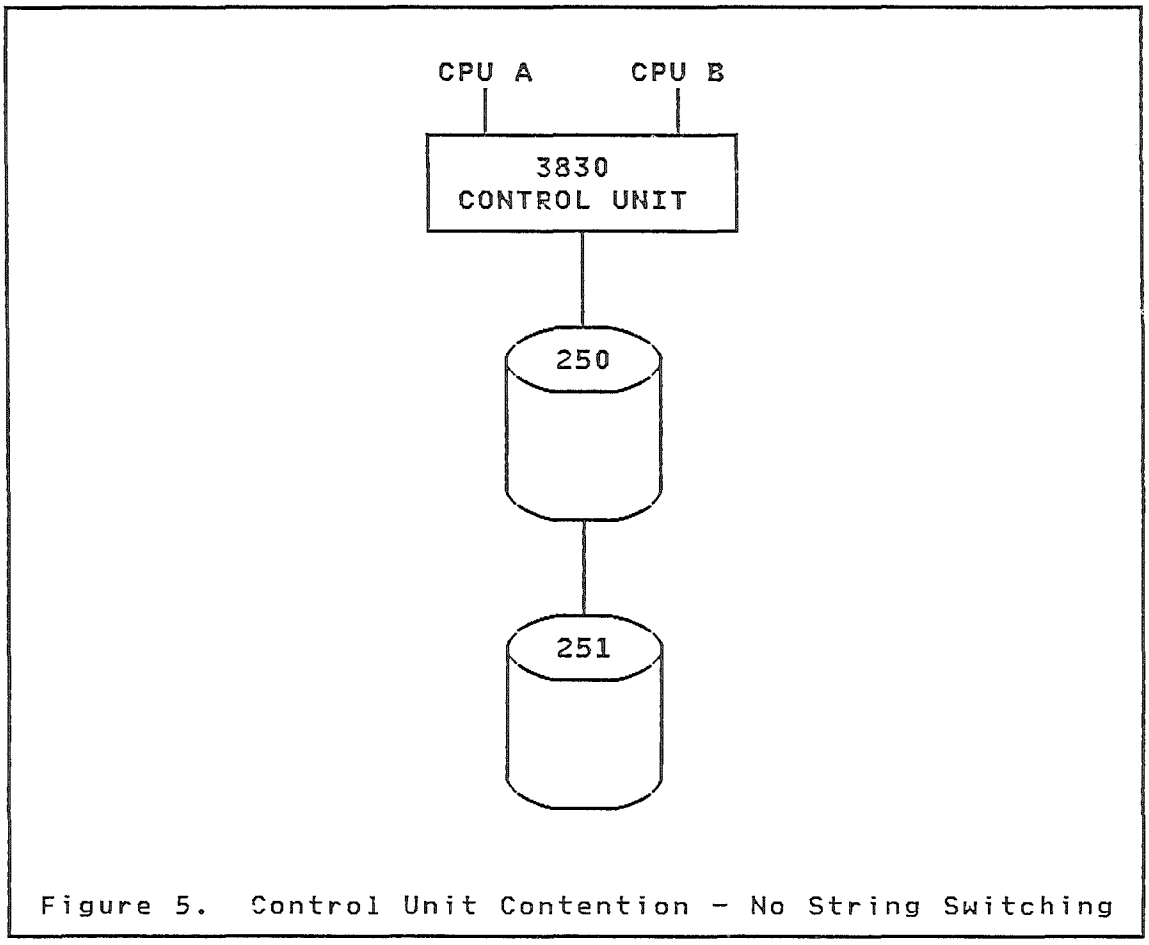


Figure 5. Control Unit Contention - No String Switching

If a successful SIO was issued to device 250 from CPU A and then CPU B attempts to issue a SIO to device 250, CPU B will get a device busy condition providing the data transfer has not yet begun from 250 to CPU A. (If the data transfer had begun CPU B would get a control unit busy). Another way CPU B would encounter a device busy condition is if CPU A had issued a RESERVE for device 250. Under this condition CPU B could not access device 250 until CPU A issues a RELEASE. Any attempts by CPU B to issue a SIO prior to the RELEASE would be rejected with a device busy condition. DADSM, Catalog Management, the Linkage Editor, JES2 MAS, and RACF are the major users of Reserve/Release. Sharing of volumes with a high rate of Reserve/Release activity should be avoided because of the potential lock-out of the other CPU.

A device busy condition cannot be encountered if the control unit is attached to only one CPU and string switching is not installed. This is because software sets the UCB BUSY bit on when a successful SIO is issued. The Busy bit remains on until the I/O operation is complete. This busy bit is an indication to the software that an I/O operation is in process and that no new I/O operations are to be started to this device until the current one completes. Therefore, if a SIO is issued and a

device busy condition is returned, it must indicate some other CPU is accessing the device.

With string switching, a 'head of string' busy condition is also indicated by the Busy bit in the CSW. Refer to Figure 6.

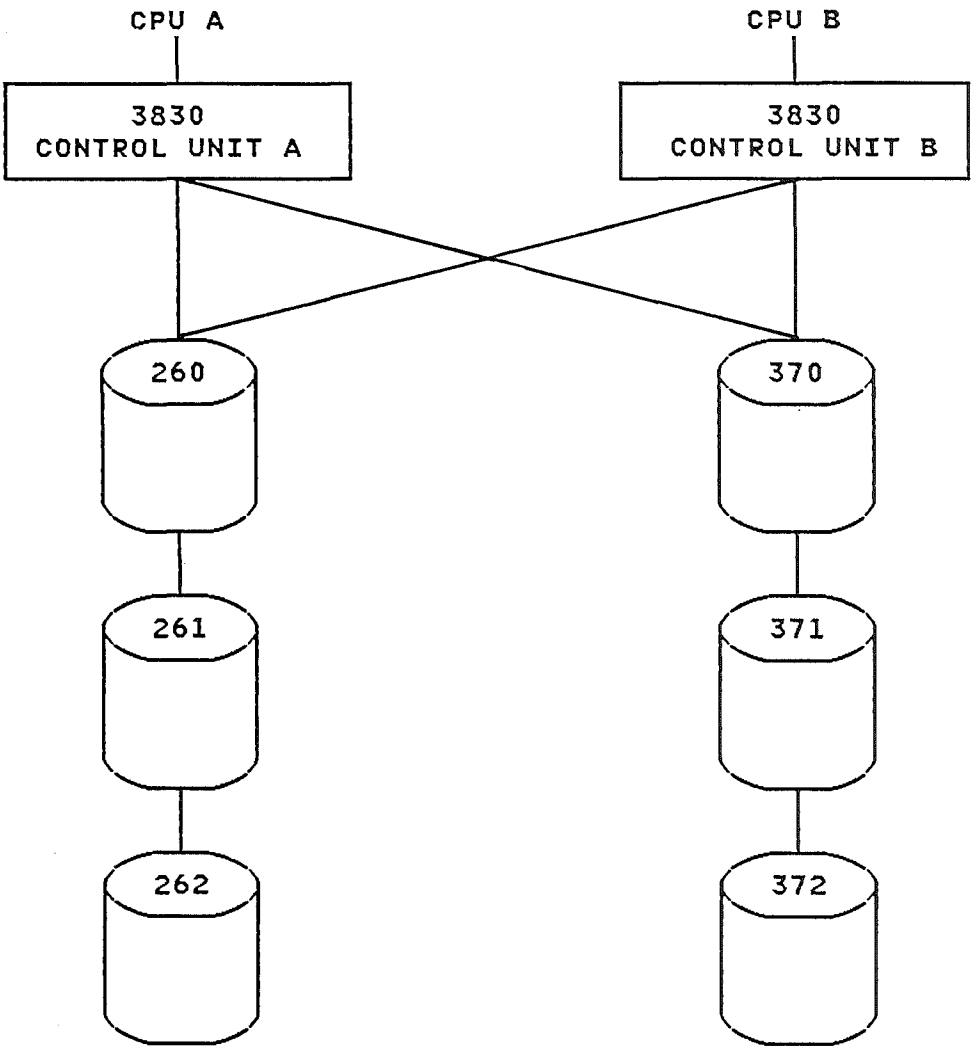


Figure 6. Control Unit Contention - String Switching

A head of string can be switched to either Control Unit A or Control Unit B but not both at the same time. A 'head of string' is busy when it or any of the devices on its string are transferring data. For example if data were being transferred from device 262 to Control Unit A and Control Unit B tried to access 261, a device busy condition would be returned to Control Unit B.

% RESV DELAY can be significant. Delays of well over of 100 milliseconds have been observed. Correcting it can involve anything from data set placement to reconfiguring the I/O subsystem.

The concept of isolation discussed in the section on Channel Configuration can also be successfully applied to shared DASD. The effects of contention caused by multiple processors accessing the same data can be difficult to manage. However, if this contention can be isolated and contained to a specific area, I/O subsystem performance can become much easier to manage. For example, spool packs, catalogs, and program libraries are generally shared among processors. Paging devices should never be shared. You wouldn't want to mix the two on the same channels and control units. Access to the shared data (i.e., spool packs, catalogs, etc.) would interfere with access to the paging devices unnecessarily. Again isolating the shared data to its own channels and control units would allow you to monitor and maintain the appropriate performance level of the shared data without effecting the performance of the non-shared data.

One suggested implementation is to assign 'I/O units' to each subsystem (i.e., system packs, TSO, IMS, Batch, etc.). Each I/O unit consists of two 3830 control units and two strings of string switched spindles. Any given subsystem may require one or more 'I/O units' depending upon performance requirements, I/O rate, amount of data, etc. Only those 'I/O units' that support subsystems running on multiple processors are shared. The remaining I/O units are isolated to individual processors.

For additional information on shared DASD, refer to 'DASD Configuration and Sharing Considerations' (GG22-9052).

In the real world, isolation is probably not 100% attainable. However, it should be a major objective in designing an I/O configuration. Without the systematic approach it suggests, I/O subsystem performance may become unmanageable.

ARM CONTENTION

Arm contention is caused by two or more active data sets on the same volume and/or long SEEK distances. The effects of arm contention can appear in queue time and service time. Service time is increased when the arm is required to move long distances from one data set to the next. Generally, any non-paging volume (paging configurations are discussed later) with an average seek distance of more than 40 to 50 cylinders should be examined.

Queue time can also be affected by arm contention. The increase in service time when long SEEKs are present increase device

busy and therefore increase the probability that the request will be queued. Several techniques for determining SEEK patterns are available. The section on GTF Data Analysis describes one approach.

Arm contention can also be caused by too many active data sets on a volume. This condition usually shows up as high device busy with a high queue time but reasonable service time.

In any case, excessive arm contention indicates some sort of data set placement activity is needed. It may require moving high activity data sets to low use volumes or reallocating data sets closer to one another to reduce arm movement.

Another condition that can cause excessive service times is volumes with large VTOCs that are searched frequently. Generally this type of volume has many small data sets. This condition may require moving some data sets to a different volume to reduce VTOC search time. The performance exposure here is not necessarily to the jobs accessing data on this volume but to other more critical applications that cannot access their data because the channel and control unit are busy doing a VTOC search.

A similar condition can occur with large program libraries and their directory searches. Situations like this may require splitting the library across several volumes.

A properly configured paging subsystem is critical to MVS performance. Because paging is a global service, it can affect the performance of everything running in the system. Therefore, special attention must be paid to its specification.

One very critical key to system performance is ensuring the right number of address spaces are in storage. Too many address spaces in storage can drive the paging rate to excessive levels. On-line and interactive subsystems, generally the most critical subsystems in the system, are usually the first to feel the impact of an excessive paging rate. So the goal of the system tuner should be to ensure the paging rate is kept to a acceptable level.

The Multiprogramming level (MPL) adjustments that MVS provides is the mechanism used to keep paging reasonable. So in essence, the procedure for tuning a paging configuration is a two step process. First, control the MPL level to ensure paging does not exceed acceptable levels. Second, define a paging configuration to handle the paging rate dictated by the MPL level.

Let's discuss controlling the MPL first. Because there are some differences between the way an MVS/SE and SU7 (non-MVS/SE) system control MPLs, each system will be addressed separately.

SU7 SYSTEM

Experience clearly indicates one of the better ways to control MPL adjustment is via page fault rate. However, the page fault rate (PTR) constants are essentially disabled in the SRM's MPL decision algorithm. Therefore, it is necessary to modify the SRM constants to activate PTR and deactivate the current controlling constants: ASMQ, UIC, and CPU.

The rationale for this recommendation, which is based on experience, is basically the following:

1. The ASMQ is is essentially a measure of how well the installation has defined its paging configuration. Excessive queue lengths generally indicate path or device contention problems. Throttling a processor because of an improperly defined paging configuration is somewhat akin to 'the tail wagging the dog'. The proper action is to define the paging configuration correctly, thus reducing or eliminating the ASMQ.

2. There isn't much one can tell about paging by looking a CPU utilization only. If someone were to ask you to estimate the impact of paging on a processor running at 98% utilization you would find it extremely difficult to do so without looking at other data such as paging rates. For example, it is not unusual to find a processor running at 100% where paging is insignificant. Attempting to lower the MPL under these conditions, would aggravate the problem by causing additional non-productive CPU utilization for the Swap-Out(s).

3. Experience has shown that the UIC is not always consistent enough to provide predictable MPL control over a wide variety of system loads. There are some environments where UIC works well, others were it doesn't. We do know that PTR appears to provide consistent MPL control.

One procedure for activating PTR follows:

1. Disable the ASMQ, UIC, and CPU SRM constants by setting them to the values shown in Figure 7.

	HIGH	LOW
ASMQ	100 (0064)	100 (0064)
UIC	0 (0000)	0 (0000)
CPU	100 (064F)	100 (064F)

Figure 7. SRM Constant Disabling

Note: The numbers in parentheses are the actual values, in hex, that should be stored in each SRM constant.

2. Set the SRM Page Fault Rate constants (PTR) as shown in Figure 8

PROCESSOR	PTR HIGH	PTR LOW
3031/158	25 (0019)	20 (0014)
3032/168	35 (0023)	30 (001E)
3033/168MP	55 (0037)	50 (0032)

Figure 8. SRM PTR Value Recommendations

Note: The numbers in parentheses are the actual values, in hex, that should be stored in each SRM constant.

These are initial values and may need adjustment depending upon the performance objectives of the system. Most installations determine their final PTR values by measuring page fault rate (i.e., non-swap, non-VIO page-ins plus Reclaims) against on-line or interactive response. Generally there is a correlation between page fault rate and response time. As the page fault rate increases so does the response time. The point at which the response exceeds the performance objectives defines the PTR High value. PTR Low generally should be 5 to 10 below PTR High.

Another way to determine the proper paging level is to measure page faults per second that a subsystem is encountering. For example, IMS response may begin to degrade when the IMS control region encounters more than 2 to 3 page faults per second or a message region exceeds 3 to 4 page faults per second. JES3 appears to degrade when it incurs more than 8 to 10 page faults per second. These values assume a properly tuned paging configuration.

3. Monitor utilization. The Page Data Set Average Service Time, as reported by RMF, should generally be 50 to 70 milliseconds. This is because the Auxiliary Storage Manager (ASM) attempts to build channel programs that will achieve a maximum service time of 50 milliseconds. (Page Data Set Average Service Time is calculated by dividing Page Data Set % In Use by SIOs per second.) In systems were paging is light, service time for local, PLPA, and common data sets may be less than 50 milliseconds. However, because requests to the Swap data sets generally involve the transfer of 5 to 7 pages, a service time of less than 50 milliseconds is unlikely.

   Because ASM does its own internal queuing, Q time for paging requests cannot be totally determined. In this case you can use % Device Busy as as indicator of Q time. Generally, 3330 and 3350 paging device utilization should be kept below 35% . A non-zero Average Queue Length value for a paging volume would probably indicate some other active data set on the same volume. Check the average number of OPEN data sets for the device. This value should be zero because page data sets are not counted as OPEN data sets in this calculation. % CU Delay should be minimized. % RESV Delay should always be zero. A non-zero value could indicate that this device is being shared with another processor.

   Channels servicing paging devices generally should be kept below 35% utilization. This will minimize delays caused by channel busy conditions.

   The following should also be considered when designing a paging configuration.

   a. Paging volumes should be dedicated, that is, no other active data sets on the same volume.

   b. Use 2305 for local and PLPA data sets. Heavily loaded environments with subsystems such as IMS, VTAM, JES3, etc., that make extensive use of CSA might also consider the common area on a 2305.

   c. Swap data sets are generally less sensitive to arm movement because the arm is required to move a maximum of one time to access a swap set. Therefore 3330 and 3350 devices perform very well as swap devices.

MVS/SE RELEASE 1 SYSTEM

A slightly different approach may have to be taken to control the MPL when running under MVS/SE. This is true because changes have been made to the decision algorithm. It is possible the default values of the MVS/SE algorithm may allow a demand paging rate higher than was acceptable under the SU7 System. If this is the case, there are at least two possible approaches. The first is to set the PTR values in the MVS/SE system to the same values used in the SU7 system. The second approach is to change the demand paging rate (DPR) and/or the milliseconds-per-page (MSPP) in the MVS/SE algorithm until the proper demand paging rate is achieved.

The following description of MPL control in MVS/SE will help in understanding how the algorithm constants may be modified.

The changes made in MVS/SE to the multiprogramming level control decision process are continuations of the direction set by SU7. The domain concept remains intact. What has been refined are the parameters used in deciding how or when to adjust domain multiprogramming levels (MPLs). A second set of changes in MVS/SE are new external parameters for defining domain control in the IPS.

The is the resource monitor routine of the SRM that samples various system variables and decides periodically whether this system is under, over, or correctly utilized. In SU7, this decision was made every thirty seconds; in MVS/SE it is made every twenty seconds in order to be more responsive to changing conditions in the system. In SU7, there were five variables sampled and averaged, but the decision algorithm effectively no-Oped two of them. The ones left were long term average CPU utilization (CPUA), average ASM queue length (ASMQA), and average high system UIC (UICA).

In MVS/SE, there have been changes to the over/under utilized decision process. First, a short term average CPU utilization is used for CPUA, making the SRM more sensitive to changing CPU usage. Average ASM queue length, ASMQA, while still in the decision logic, has threshold values ('happy values') of (100,100), which prevent it from having any effect on the MPL decision. In addition, the meaning of the ASMQA number is changed somewhat from SU7. (In MVS/SE, it includes pages for swap data sets, but in SU7 it did not.) The only active system contention indicator remaining intact from SU7 is UICA. Although UIC updating has changed somewhat in MVS/SE, it produces results identical to SU7 in the area where the SRM is sensitive (UICA thresholds of 2 and 4). Page fault rate (PTR) is still in the decision logic just as it was in SU7, still no-Oped. The same is true of logical channel contention.

Two new variables are used in the MPL adjustment algorithm for MVS/SE. One is the demand paging rate, that is the average num-

ber of non-swap, non-VIO page-ins and page-outs per second, both in the private area and the system pageable areas. The other variable is page delay time, or the average time needed to move a page into or out of storage. This variable is discussed in more detail below. Neither of these variables, on its own, is actively used as a measure of contention. What the SRM in MVS/SE considers bad, instead, is a high demand paging rate when it takes the system a long time to transfer a page into or out of storage. Similarly, it is bad if the demand paging rate is high and the CPU utilization is high, for then the paging overhead then is potentially taking the CPU away from some useful element of work. This view leads to the new factor in the MPL decision process - a combination of demand paging rate with average CPU utilization and page delay time. Therefore, in the new SRM algorithm, the system is over utilized and the system MPL should be decreased if:

```
        UICA < 2

  or    CPUA > 100.9%

  or    (demand page rate > HIGH and
        (CPUA > 98% or page delay > 130 ms.))

  or    PTR > 100 *

  or    Page delay > 1000 ms *

  or    ASMQA > 100 *

  or    logical channels over utilized and CPUA < 98% *
```

Figure 9. SRM Algorithm - Determining An Over Utilized System

Note: * indicates threshold value effectively eliminates this test.

The system is under utilized and the system MPL may be increased if:

```
        UICA > 4

   and  CPUA < 98% (a threshold change from SU7)

   and  (demand page rate < LOW or
        (CPUA < 95% and (page delay < 100 ms.))

   and  PTR < 100 *

   and  Page delay < 1000 ms *

   and  ASMQA < 100 *

   and  logical channels are not over utilized.*
```

Figure 10. SRM Algorithm - Determining An Under Utilized System

Note: * indicates threshold value effectively eliminates this test.

The low and high demand paging rate (DPR) thresholds are CPU dependent. These thresholds are:

CPU	158	168	3031	3032	3033
DPR Low	17	42	20	46	70
DPR High	22	52	26	57	88

Figure 11. Demand Page Rate (DPR) Thresholds: For an AP or MP, The values are multiplied by 1.7.

To restate the intent of the new logic:

1. A low demand paging rate (<LOW) indicates possible system under utilization (regardless of CPU or page delay).

2. Demand paging rate may be ignored for system under/over utilization, if CPU utilization is low and paging operations are accomplished in a 'short' time.

3. A high demand paging rate (>HIGH) is bad (system over utilized) only if CPU utilization is also high or it takes a 'long' time to perform paging operations.

Once the decision is made whether to increase, decrease, or maintain the system MPL, the question becomes - which domains? As in SU7, if the system MPL must be lowered, the domain with the lowest contention index that is not already at its minimum MPL has its target MPL lowered by one. If the system MPL should be raised one, the domain with the highest contention index, not already at its maximum MPL, has its target MPL raised by 1. And if the system MPL is to remain steady, contention in the system is balanced by raising one domain's target and lowering another's. In MVS/SE, the new domain controls are alternative means for calculating a domain's contention index. Since these controls are new external parameters, they are fully discussed in the MVS Initialization and Tuning Guide.

The simplest form of the new formats for specifying domain contention is the fixed contention index (FWKL). It provides a method of defining a fixed level of contention for a domain, for example, a very high level for first period TSO. It also provides a simple specification for domains where there is no swapping, either because of a high minimum MPL or because all the members are non-swappable.

The other two new methods of domain control use the service units accumulated by all users in the domain in the calculation of the contention index. In order to support these new calculations , the workload manager keeps track of total domain service. Every twenty seconds, the service rate for each domain over the last twenty seconds is calculated, then averaged with the previous value of the domain time-weighted service rate. This becomes the new time-weighted service rate, which is the key variable for each domain. This variable is reported in the DISPLAY DOMAIN operator command, and is traceable with RMF. It is simply an average service rate for the domain over many previous intervals, with the previous twenty second interval being more significant than the next previous, and so on. For a DOBJ domain, this number is used with the specified domain objective to arrive at a contention index.

For an AOBJ domain, the time-weighted service rate for the domain is divided by the average number of ready users in the domain. The average service per ready user is used with the specified AOBJ domain objective to arrive at a contention index. This ready-user-average (RUA) is also time-weighted in MVS/SE (as it was in SU7), and is maintained internally as sixteen times the average number of ready users, for more precision than whole numbers allow. When the RUA is reported via DISPLAY DOMAIN or used in contention index calculations (AOBJ or Weight), it is divided by sixteen and rounded to the nearest whole number. When traced with RMF, however, RUA shows sixteen times the average number of users. While domain service includes all service in the domain, including that going to non-swappables, the RUA is a count of the average number of ready, swappable users in the domain.

Another change in the MPL target adjustment is that some allowance is made for a domain that may have a highly variable number of users ready over any twenty second interval, such as a TSO domain. In SU7 the domain's target MPL was lowered by one if it was determined that there was not enough ready work in the domain to justify its current target. Not enough work meant that the average number of ready users (RUA) fell more than one below the current target MPL. In MVS/SE, the maximum number of ready users encountered over a sampling interval is also maintained. Not enough work in a domain means, in MVS /SE, that the current target MPL is more than one above the halfway point between the newly calculated RUA and the maximum number of ready users over the last interval. This change should allow a dynamic domain to maintain a higher target MPL to deal with instantaneous peaks in the number of ready users.

Page Delay Time

One of the new factors in the MPL adjustment algorithm is called page delay time. This section is a short explanation of what this variable is and how it is calculated.

As explained above, the logic of the MPL target adjustment process considers a high demand paging rate to be bad if it takes a long time to do a paging operation. The first problem is how to define this length of time and how to measure it. Consider first a demand page fault, that is a job is running and references some data and causes a page fault which requires I/O. The amount of time this job is delayed because of the page fault is the time from the page fault until the task (or SRB) is ready to run again after the page has been brought into storage. This delay may be critical for some types of work, when looked at in conjunction with the number of page faults the task encounters. If a task must receive a certain amount of service per second in order to accomplish its work, the time spent waiting for page faults can be an important factor. If, for example, an IMS control region or CICS memory has to receive 500 milliseconds (ms) of CPU service per second in order to handle 'n' transactions per second, then suffering 10 page faults per second, each of which took 60 ms to resolve, would make the task incapable of handling all its work. This example ignores higher priority work and waits for other I/O, so even 5 page faults per second (at 60 ms each) could be a problem if the task had to wait for 10 normal I/O operations each second (at 30 ms each) as well as get 500 ms of CPU each second in order to process its work.

The page delay factor, then, is a recognition of the importance of this delay due to paging. The time interval referred to in this example, however, is not measured during normal system operation. The amount of time a paging request is under the control of IOS is measured in RMF for MVS/SE as the average ser-

vice time of each paging data set. It is this time which corresponds to ASM's 50 ms 'burst' value, but note that it does not take account of the time a request may wait on ASM's internal queues before going to IOS. An approximation of the total time to satisfy a page fault is the length of time the paging request is on ASM's internal queue, for the request is queued almost immediately after the page fault occurs, and the request is not taken off the queue until the I/O is complete, just before the task is made ready to run.

This led to the use of ASM's queue as the place to look for page delay time. The average number of requests on the ASM queue, (that is, page requests waiting to be processed or with I/O active), is calculated by the SRM as ASMQA. In SU7 it was used as a contention indicator in the MPL decision algorithm. Using a relationship known as Little's Law, it can be proven that the average length of time a request spends on the ASM queue is the average length of the queue (ASMQA) divided by the total rate of requests. This is the same law that is used when we calculate device service times from RMF reports, using the average device queue length to calculate that portion of the service time which a request spent on the device queue. The law remains valid for the ASM calculation in spite of the differences - that time on the ASM queue represents time truly waiting plus the time the I/O is active, and the fact that ASM uses many different devices to service its requests.

The total rate of paging requests processed, referred to above, is simply the total paging rate, which the SRM now also keeps track of (traceable in RMF as RCVPAGRT). To make the calculation valid, the ASMQA number should represent the average of all requests queued, since we will divide by the rate for all requests processed. In SU7, ASMQA was the average number of requests queued to common or local paging data sets, but not to swap data sets. In MVS/SE, ASMQA is the average number of requests queued, including those for swap data sets. (Note that this means an installation with swap data sets which used ASMQA in the SU7 MPL adjustment logic should not set the ASMQA thresholds blindly to (7,10), or whatever the installation used with SU7, in an effort to duplicate the algorithm for SU7; the ASMQA values seen in an MVS/SE system are likely to be higher.)

All of the key elements are now in place. The average length of time needed to process a paging request is the result of dividing the average length of the ASM queue (ASMQA) by the rate of requests processed, or the total paging rate (RCVPAGRT). This is the average length of time needed to process <u>any</u> paging request, that is, page-out, page-in, VIO, swap, non-swap, common, or local. This number is given the name RCVMSPP (for milliseconds per page), and is in units of milliseconds. It is not the average length of time a task is delayed by a page fault, although the intent is to reflect that. RCVMSPP is traceable with RMF.

The second problem in using page delay in the MPL algorithm, now that it is defined, is in assigning thresholds. Based on samples of measurement runs, the default values have been set to (100,130) in MVS/SE. When looking at these values it is important to remember that MSPP is not the time a task is delayed due to a page fault, for 100 ms would be unacceptable, but it is a reflection of how long all paging requests take to be satisfied, on average.

The usefullness of milliseconds-per-page (MSPP) in controlling an installation's MPL will depend to a large degree on the characteristics of its paging subsystem and paging in the system. MSPP is sensitive to the paging rate and the paging configuration. More paging devices, or faster paging devices, tend to produce a lower MSPP. Higher paging rates (total paging) tend to increase the MSPP, even if the paging configuration is adequate to handle the load. In a system over configured for paging, the MSPP should never reach 100; if there were any storage constraint, the UICA thresholds would control the MPL before MSPP approached its threshold. In a heavy swapping TSO system, the MSPP could get high, but the demand paging rate that it is coupled with would probably not get high. These demand paging rate thresholds were chosen such that the low threshold is a paging rate that consumes about 5-6% of the processor.

MVS/SE RELEASE 2 SYSTEM

With the availability of MVS/SE2, a new factor has been introduced into the paging configuration tuning process. Prior to MVS/SE2, the objective of tuning a paging configuration was to keep the system paging rate at a level that would not cause excessive page faulting within the critical subsystems. MVS/SE2 now provides the capability of isolating storage to critical subsystems so that they are allocated the real storage frames they require for performance.

A description of the Storage Isolation Function and recommendations for its use are described in the Washington Systems Center Technical Bulletin, 'MVS System Extensions Release 2.0 - Planning Information' (GG22-9028). The following is an excerpt from that document and highlights the recommendations for use of the Storage Isolation Function.

Pageable frame stealing is the process of taking an assigned frame away from an application or the common area and making it available for other purposes. When the demand for pageable frames exceeds the supply of available frames, the SRM will steal those frames which have gone unreferenced for the longest time and return them to the system.

Each application has critical frames which if stolen will impact the performance or throughput of the application. The decision on which frames in the system are stolen first is based on which frames have gone the longest period without being referenced. In most cases, the time between each reference to a specific page is based on the existing workload. When the application is processing a medium or light workload the critical frames have a higher probability of being stolen. This problem also occurs in memory constrained environments in which all frames appear as if they had been recently referenced. It is believed that within the application the critical frames do tend to be referenced more often, however, in both the lightly loaded situation and in the storage constrained system the critical frames are still stolen.

In MVS/SE R2.0, SRM provides a facility which allows for the storage isolation of applications and the common area in both the storage constrained and non-storage constrained environments. The algorithm's control is based on the manipulation of the working set size. An application's working set consists of all the LSQA and local frames allocated to it. The working set for common consists of all the frames allocated from the common area.

Using the information specified in the IPS, the algorithm calculates a target working set size for each application and for the common area. SRM will not request the stealing of frames from an application as long as the application's working set size is less than or equal to the target working set size. SRM

will also not request the stealing of frames from the common area as long as the common area working set size is less than or equal to the target working set size for common.

An installation may place limits on the 'Least Recently Used' steal algorithm. In tailoring these limits, an installation may base their control on several factors.

An application may require a minimum number of frames allocated to it for acceptable performance. The application's execution characteristics may be such that this requirement for frames need not change as the load on the application changes. The installation may specify this requirement in the IPS.

An installation may not be able to achieve acceptable performance for an application, by specifying a static minimum working set size. The performance of the application may be dependent on the application's page-in rate. The installation may specify this requirement in the IPS.

An application may always require a minimum number of frames for acceptable performance but this number may not be static and as the load on the application changes so does the required working set size. This requirement can also be specified in the IPS.

The following discussion is included to create a basis for an understanding of how to determine realistic values for the minimum and maximum page-in rates to avoid either extreme underprotection or overprotection of private address spaces for which page-in storage isolation is specified.

The first thing to notice is that the page-in rate used in storage isolation is calculated differently for private address spaces and for common area.

Private Address Spaces

$$\text{PAGE-IN RATE} = \frac{\text{\# NON SWAP + NON VIO PAGE-IN REQUESTS}}{\text{ACCUMULATED EXECUTION TIME SECONDS}}$$

Figure 12. Storage Isolation Page-In Rate Calculation For Private AddressSpaces.

The page-in rate is recalculated every time the two requirements listed below have been satisfied.

1. At least 10 seconds of residency time has elapsed since last calculation.

2. At least 1 SRM second of CPU time has been executed.

Common Area

$$\text{PAGE-IN RATE} = \frac{\text{\# PAGE-IN REQUESTS}}{\text{ACCUMULATED ELAPSED TIME SECONDS}}$$

Figure 13. Storage Isolation Page-In Rate Calculation For Common Area.

The page-in rate is calculated if at least 10 seconds has elapsed since last calculation.

The reason for using execution time and not elapsed time in the page-in rate calculation for private address spaces is, that it may take a long time for an address space to use 1 SRM second of CPU time when the application load is low. This could result in an extremely low page-in rate if it were calculated based on elapsed time, even when the application in reality experienced a very high page fault rate in the short intervals where it actually did execute transactions. The page-in rate would easily be less than 1, the minimum value which can be specified for page-in rates. The target working set would therefore either not change or constantly be reduced resulting in no protection of the working set and poor performance for the application.

A similar problem does not occur with the page-in rate for common area, where there is no need for an execution time threshold because of the relatively constant activity in this area even in lightly loaded systems. The page-in rate which is calculated every 10 seconds of elapsed time results therefore in relatively significant page-in rate values if just some paging activity has occurred.

There is a risk for overprotection of private address spaces with page-in rate storage isolation. Very small values for minimum and maximum page-in rates may result in unnecessary performance degradation of non-isolated applications. Inversely, there is a risk for underprotection if too high page-in thresholds are specified, resulting in poor performance for the page-in isolated address space (but this risk is probably less).

The following considerations outline a methodology for determining the page-in rate thresholds for the IPS specifications and identify the necessary information to make this determination.

1. Set objectives for acceptable minimum and maximum average response times for transactions handled by a given application (address space).

   a. Minimum acceptable response time = RTmin

   b. Maximum acceptable response time = RTmax

2. Determine the average value of the 'response time deltas' for the different components, which make up the average total response time of a transaction.

   a. Average CPU seconds needed per transaction (Rcpu)

   b. Average I/O wait time per transaction (Rio)

   c. Average time delay per transaction caused by higher priority work (Rint)

3. Determine the average time it takes to resolve a page fault. A reasonable good value for that is the average page delay time also called MSPP, which can be traced with RMF.

4. Determine the average minimum (MINdelay) and maximum (MAXdelay) time a transaction may be delayed because of page faults - insuring that the min/max response time objectives can be satisfied.

```
 - (Rcpu+Rio+Rint)
MAXdelay = RTmax - (Rcpu+Rio+Rint)
```

Figure 14. Storage Isolation - MIN/MAX Delay Calculations

5. Determine the minimum and maximum allowable page fault rate (page-in rate) per transaction - insuring that the min/max response time objectives can be satisfied.

$$\text{MIN PAGE-IN/Trans} = \frac{\text{MINdelay}}{\text{MSPP}} \text{ per transaction}$$

$$\text{MAX PAGE-IN/Trans} = \frac{\text{MAXdelay}}{\text{MSPP}} \text{ per transaction}$$

Figure 15. Storage Isolation - MIN/MAX Page-In Calculations

6. Determine the average number of transactions which will complete per executed SRM second.

   This is done because the page-in rate for private address spaces only will be calculated every time 1 SRM second of CPU time has been consumed.

$$\text{Ntrans} = \frac{\text{1 SRM SECOND}}{\text{Rcpu}}$$

Figure 16. Storage Isolation - Transaction Rate Calculation

7. Determine the page-in thresholds to be specified in the IPS for address spaces with page-in storage isolation, insuring that the min/max response time objectives can be satisfied.

$$\text{MIN PAGE-IN} = \text{Ntrans} * \text{MIN PAGE-IN/Trans}$$

$$\text{MAX PAGE-IN} = \text{Ntrans} * \text{MAX PAGE-IN/Trans}$$

Figure 17. Storage Isolation - Threshold Calculation

8. Installations which plan to implement this kind of storage isolation control should begin to collect the needed information now (if it is not already available) via application LOG tapes, performance monitors, RMF, etc.

A less precise method for setting the page-in thresholds is to correlate a user specified performance variable against page-in rate. For example, an IMS shop has established a response time objective that 90% of the transactions must complete within 3-5 seconds. A representative period is selected to collect data. The IMS log contains the transaction response times. An IMS reduction program gives the 90 percentile response times for each sample interval during the measurement period. RMF Monitor II Background Session with the ARD report active provides the page-in rate per CPU second for the IMS control region. The RMF sample time (SINTV) was chosen to equal the IMS sample interval. The data is plotted in ascending response time order, where the vertical axis is the response time, and the horizontal axis is the page-in rate per CPU second. The page-in rate where the 90 percentile response time crosses the 5 second objective is specified as the high threshold. The page-in rate where the 90 percentile response time crosses the 3 second objective is specified as the low threshold.

RMF Version 2 Release 2, MVS/SE R2.0 Feature will support storage isolation in both foreground and background reports.

ARD Report

1. The target working set size (new field).
2. The page-in rate per CPU second (new field).
3. The long term average storage allocation (old field).

ASD Report

1. The actual number of allocated frames (old field). This field should be used to determine working set size.

SPAG Report

1. The target working set for common area (new field).

RMF Tracing

RMF will support tracing of some of the new storage isolation variables.

Recommendations For Use Of Storage Isolation

1. WORKING SET ISOLATION only.

   This is the most simple form of storage isolation control. It may be used to protect the working set for applications/common area with a very well defined and static working set.

   The minimum working set (MIN WSS) should be specified a little above the critical working set (OVERPROTECTION), since the target working set (TWSS) in this situation is always equal to MIN WSS and no adjustment is done to the target.

   Note: Allocated pages above the specified maximum working set (MAX WSS) may be exposed for extreme stealing. MAX WSS should therefore be given a high value (MAX WSS=32767), unless the trimming is a wanted effect.

   A low MAX WSS can be used to limit the number of frames allocated to an application. One should however be careful not to overuse this technique for swappable address spaces, since storage isolated pages are not considered in the determination of the average high system UIC. The effect may therefore be a 'distortion' of the SRM control algorithms, which are dependent on this UIC value.

2. PAGE-IN RATE ISOLATION only.

   This form of storage isolation may be used to protect the working set for applications/common area where the working set at any given time is constant (static), but where it also may vary over time caused by load variations.

   Page-in, however, has some related problems which may or may not be serious.

   a. The TWSS can here vary between 0 and 32767. TWSS will be initialized to 0 (IPL and SET IPS) leaving the isolated address spaces/common relatively exposed for normal stealing until a reasonable 'good' TWSS has been established.

b. Page-in rate isolation should not be used to protect applications which constantly are able to generate a high page-in rate independent of the number of allocated frames, since that may result in steadily growing working sets beyond the wanted/necessary level.

3. PAGE-IN RATE and WORKING SET ISOLATION.

This is probably the most useful form of storage isolation for protection of the working sets for applications/common area in real life situations.

TWSS can here vary between MIN WSS and MAX WSS based on the actual page-in rate in relation to the page-in thresholds.

This technique provides a lower limit protection of the TWSS via the MIN WSS specification. The lower limit (MIN WSS) should be specified below the required (critical) working set if such can be determined (UNDERPROTECTION).

The page-in rate specifications provide the mechanism to adjust the target working set to variations in the critical working set size caused by load variations over the day/week/month/year or new application development.

Note: An address space may be exposed to extreme stealing if the number of allocated frames exceeds the maximum allowable allocated frames, MAX WSS (See Section 5.3 for details). MAX WSS should therefore be given a reasonably high value (maximum is 32767), unless the stealing (trimming of WSS down to MAX WSS) is a wanted effect, in which case the MAX WSS may be used as an effective upper limit for the working set for a given address space.

THIS PAGE INTENTIONALLY LEFT BLANK

Having discussed the various 'rules of thumb' associated with DASD tuning, a look at an example taken from actual measurement data and a step through the analysis process seems appropriate. In the following examples the assumption is that RMF will provide our initial source of data. Depending upon what the RMF data indicates, additional data may have to be collected to further isolate any problems. Also note that the RMF and GTF data that follows does not necessarily depict an actual performance problem. This data has been included here to help illustrate an approach to data analysis.

There are several approaches one can take when analyzing I/O problems. One approach, which appears to work well, is to start at the source; the device. The key to the analysis, as was described earlier, is response time. response time is comprised of two major components: queue time and service time.

RMF DATA ANALYSIS

The first pass at analyzing an I/O problem should be an examination of RMF's Direct Access Device Activity Report. Pay particular attention to Average Queue Length and Average Service Time.[5] Make note of any device that violates the 'rules of thumb'. For these selected devices, calculate the response time (queue time plus service time). This process will provide an initial priority list of devices needing attention. Usually the device with the longest response time should be addressed first.

To illustrate this process, Figure 18 on page 46 contains a sample RMF Device Activity Report. Note that RMF reports service time in a field called Average Service Time. To save space only the relevant RMF data is shown.

---

[5] Average Service Time is available only on the MVS/SE version of RMF. However, it is easily calculated by dividing % Device Busy by Device Activity Rate.

RMF DEVICE ACTIVITY REPORT

DEV ADR	VOLUME SERIAL	DEVICE ACTIVITY RATE	AVG SERV TIME	AVG QUEUE LGN	% DEV BUSY	% CU DELAY	% RESV DELAY
284	PAGE01	4.840	.043	.00	21.26	0.00	0.00
285	SWAP01	1.098	.048	.00	5.39	0.00	0.00
370	PAGE12	4.791	.041	.00	19.92	0.41	0.00
371	SWAP11	.961	.050	.01	4.98	0.21	0.00
480	P59WK1	10.497	.057	.68	60.79	1.04	0.00
481	P59WK2	15.308	.049	.57	75.93	0.62	0.00
482	SPOOL1	3.866	.059	.09	22.41	0.41	0.00
484	PAGE02	3.697	.070	.01	25.93	0.41	0.00
4A0	P59WK3	15.747	.055	1.00	86.93	0.00	0.00
4A1	P59WK4	8.483	.053	.23	45.64	2.28	0.00
4A4	PAGE03	3.697	.080	.02	29.67	1.45	0.00
4A5	SWAP02	.949	.059	.00	5.60	0.41	0.00

Figure 18. RMF Data - Device Activity Report

RMF CHANNEL UTILIZATION

CHAN ADR	PERCENT BUSY	ACTIVITY PER SECOND	AVERAGE SERVICE TIME
2	17.63	10.186	0.017
3	22.41	16.087	0.013
4	52.70	45.174	0.011

Figure 19. RMF Data - Channel Utilization

An analysis of the RMF Report is shown in Figure 20 on page 47. Only the devices that appear to require some attention are listed. Note all times are in seconds and that the devices are

listed by response time with the highest response time first. This provides a priority list of devices requiring attention.

RESPONSE TIME CALCULATIONS

DEV ADDR	I/O RATE	RESPONSE TIME	QUEUE TIME	CU DELAY	RESV DELAY	SERVICE TIME
480	10.497	.122	.065	.001	.000	.057
4A0	15.747	.119	.064	.000	.000	.055
4A4	3.697	.085	.005	.004	.000	.080
481	15.308	.086	.037	.000	.000	.049
482	3.866	.082	.023	.001	.000	.059
4A1	8.483	.080	.027	.003	.000	.053
484	3.697	.073	.003	.001	.000	.070

Figure 20. Response Time Calculations

After reviewing the data in Figure 20, several observations can be made.

Observation #1

Except for device 482 (SPOOL1) all the volumes are paging devices or work packs (P59WKX). Also note that there appears to be no contention in control unit 28X (% CU Delay = 0) and a slight amount of contention in control unit 37X.

Observation #2

The response time on the work packs appears to be a problem particularly on devices 480 and 4A0. On these devices the queue time is longer than the service time and there is essentially no control unit or shared DASD/string head contention. The assumption can be made that the queue time is caused by device busy (UCB BUSY bit on). Without going much further, it would be possible to conclude there are an insufficient number of work packs. Judging from the queue time, it would appear a minimum of two additional work packs are required.

Observation #3

There appears to be a high level of contention in control units 48X and 4AX. Because shared DASD does not seem to be involved (% RESV DELAY = 0 in all cases), control unit busy may be the cause. Device 4A0 appears to be an example of the dominant volume situation described earlier. Notice 4A0 shows no % CU delay while the other devices on that control unit are showing delays.

Having already determined additional work spindles are needed, it's now obvious that these packs should not be added to either the 48X or 4AX control units. The best candidate appears to be control unit 28X with 37X as the second choice. 28X shows no contention and Channel 2 utilization is only 17.63% .

Observation #4

Channel 4 utilization is excessive at 52.70% . However, adding two work packs to channel 2 may solve the channel 4 utilization problem, but additional measurements are needed.

Observation #5

Service time for the SWAP data set on 4A5 is higher than the Swap data sets on 285 and 371. Refer to Figure 21. What we have here is a good example of what happens to service time when contention increases.

DEV ADR	AVG SERV TIME	% CU DELAY	% CHAN BUSY
285	.048	0.00	17.63
371	.050	0.21	22.41
4A5	.059	0.41	52.70

Figure 21. Contention And Its Effect on Service Time

Note that as control unit and channel contention increase so does the service time.

Conclusions/Recommendations

The source of the problem seems to be the excessive utilization of the work packs. This appears to be causing control unit contention and excessively high channel utilization resulting in unacceptably high response times for the DASD devices on channel 4.

An action plan to solve this condition might include the following steps:

1. Add two work packs to control unit 28X.

2. Measure the system. Pay particular attention to response time on the work packs and channel 4 utilization.

GTF DATA ANALYSIS

In the previous example, it was possible to determine a course of action by analyzing only RMF data. This is not always the case. In some instances RMF may not provide the detailed information that is required. For example, take the case where channel and device utilization are reasonable and yet device service time is excessive. In this situation arm and/or control unit contention may be the cause. A condition code and seek analysis are appropriate.

One of the best sources of this kind of data is a GTF reduction program called GTFPARS (PROG# 5798-CQQ). Among other things, GTFPARS provides an excellent summary of system I/O activity. Included are minimum, maximum, and average service times, condition code analysis and seek analysis.

When using any GTF reduction program, the most important consideration is to ensure the data is collected when the problem is occurring on the system. This is generally during a peak load time. Experience indicates that one reel of GTF data written at 6250 BPI is sufficient for at least an initial analysis. On a 3032 or 168 it usually takes about 15 minutes to collect a reel of data.

A sample GTFPARS I/O Summary output is shown in Figure 22 on page 50 and Figure 23 on page 51. Assuming the report was produced because some kind of contention problem is suspected, an analysis of the data is appropriate. Note this data was collected from a non-MVS/SE system. No shared DASD was involved.

*** SYSTEM SUMMARY ***

START 10/29/76 18.26.28 STOP 10/29/76 18.45.36 ELAPSED 19.08

EXCP-SIO-IO SUMMARY

DEV	PCT BSY	EXCP COUNT	ERREXCP MIN TIME	EXCPVR MAX TIME	SIO CC0 AVG TIME	DV BUSY STD TIME	CU BUSY	CH BUSY	DEV NOP	PCI	IO	IO-DE
200		0	0	0	0	0	0	0	0	0	2607	0
250	4.88	2	0	0	1494	156	201	5	0	0	0	1494
* EXCP - SIO *		2	2100	2256	2178	110						
EXCP - SIO CC0		2	2100	2256	2178	110						
*EXCP - IO(DE) *		2	10587	11843	11215	888						
* SIO - IO(DE) *		1494	639	323901	37507	38646						
252	32.65	13206	0	0	14162	60	296	24	0	0	0	14162
* EXCP - SIO *		13206	1241	5099620	22519	65603						
EXCP - SIO CC0		13206	1241	5099620	22959	65933						
*EXCP - IO(DE) *		13206	3215	5133410	49450	68909						
* SIO - IO(DE) *		14162	1314	196282	26475	15894						
253	1.26	311	0	0	347	15	32	1	0	0	0	347
* EXCP - SIO *		311	1668	97219	5747	11990						
EXCP - SIO CC0		311	1746	97219	6823	13232						
*EXCP - IO(DE) *		311	4605	269495	51224	25884						
* SIO - IO(DE) *		347	2761	204213	41988	20028						
254	4.53	2	0	0	1379	149	185	6	0	0	0	1379
* EXCP - SIO *		2	1976	1994	1985	12						
EXCP - SIO CC0		2	1976	1994	1985	12						
*EXCP - IO(DE) *		2	14808	22623	18715	5526						
* SIO - IO(DE) *		1379	6207	294876	37780	37948						
255	20.43	4952	0	2052	8141	698	858	28	0	0	0	8141
* EXCP - SIO *		7004	1631	4261056	24549	66055						
EXCP - SIO CC0		7004	1727	4261056	26344	67041						
*EXCP - IO(DE) *		7004	3852	4294718	55750	76790						
* SIO - IO(DE) *		8140	872	251903	28822	25137						
256	6.90	2367	0	0	2658	225	271	8	0	0	0	2858
* EXCP - SIO *		2367	1749	2751110	16197	92811						
EXCP - SIO CC0		2367	1822	2751110	17441	92896						
*EXCP - IO(DE) *		2367	8201	2778649	45988	97033						
* SIO - IO(DE) *		2858	6047	262458	27754	23138						
257	4.70	2	0	0	1667	208	246	6	0	0	0	1667
* EXCP - SIO *		2	1978	1993	1985	10						
EXCP - SIO CC0		2	1978	1993	1985	10						
*EXCP - IO(DE) *		2	14832	26035	20433	7921						
* SIO - IO(DE) *		1667	6139	279936	32385	27897						
258		0	0	0	0	0	0	0	0	0	372	0
300		0	0	0	0	0	0	0	0	0	1517	0
370	24.44	4168	0	3902	8510	115	45	24	0	0	0	8510
* EXCP - SIO *		8070	1576	3406528	24882	89974						
EXCP - SIO CC0		8070	1576	3406528	25135	90208						

Figure 22. GTFPARS Output - Page 1 of 2.

*** SYSTEM SUMMARY ***

START 10/29/76 18.26.28 STOP 10/29/76 18.45.36 ELAPSED 19.08

EXCP-SIO-IO SUMMARY

DEV	PCT BSY	EXCP COUNT	ERREXCP MIN TIME	EXCPVR MAX TIME	SIO CC0 AVG TIME	DV BUSY STD TIME	CU BUSY	CH BUSY	DEV NOP	PCI	IO	IO-DE
*EXCP - IO(DE) *		8070	4041	3440733	57519	98184						
* SIO - IO(DE) *		8510	570	233765	32984	24789						
372	11.65	2593	0	0	3057	90	149	8	0	0	0	3057
* EXCP - SIO *		2593	1649	4229067	38734	184729						
EXCP - SIO CC0		2593	1745	4229067	39459	184909						
*EXCP - IO(DE) *		2593	3294	4312715	82397	189838						
* SIO - IO(DE) *		3057	1338	276655	43765	26846						
373	0.91	2	0	0	401	3	6	1	0	0	0	401
* EXCP - SIO *		2	1953	1996	1974	30						
EXCP - SIO CC0		2	1953	1996	1974	30						
*EXCP - IO(DE) *		2	25004	48321	36662	16487						
* SIO - IO(DE) *		401	2440	238733	26266	24707						
374	4.73	2	0	0	1562	66	86	3	0	0	0	1561
* EXCP - SIO *		2	1987	1993	1990	4						
EXCP - SIO CC0		2	1987	1993	1990	4						
*EXCP - IO(DE) *		2	14758	17790	16274	2143						
* SIO - IO(DE) *		1561	6124	364534	34807	32974						
375	34.64	8116	0	1459	17090	310	410	18	0	0	0	17090
* EXCP - SIO *		9575	1640	4197026	19510	63986						
EXCP - SIO CC0		9575	1640	4197026	19840	63994						
*EXCP - IO(DE) *		9575	3660	4231598	41465	69615						
* SIO - IO(DE) *		17090	1257	773549	23273	19162						
376	0.68	2	0	0	151	1	2	0	0	0	0	151
* EXCP - SIO *		2	1991	2004	1997	9						
EXCP - SIO CC0		2	1991	2004	1997	9						
*EXCP - IO(DE) *		2	14816	44387	29601	20909						
* SIO - IO(DE) *		151	12825	201224	51995	21716						
378		0	0	0	0	0	0	0	0	0	54	0
482		0	0	0	0	0	0	0	0	0	0	1
***TOTALS		38793	0	7413	61060	2096	2787	132	0	0	4798	61061

SEEK DISTANCE SUMMARY

DEV	COUNT	MIN	MAX	AVG	STD	DEV	COUNT	MIN	MAX	AVG	STD	DEV	COUNT	MIN	MAX	AVG	STD
250	1489	0	3	0.34	0.59	252	14158	0	70	3.47	11.64	253	343	0	332	114.22	140.57
254	1375	0	4	0.48	0.86	255	8137	0	227	35.37	74.99	256	2854	0	69	8.78	13.73
257	1663	0	22	4.53	6.19	370	8413	0	163	23.55	42.39	372	3053	0	103	15.55	20.60
373	397	0	201	1.76	14.19	374	1558	0	3	0.37	0.61	375	17086	0	275	23.88	56.99
376	147	0	2	0.12	0.34												
***TOTALS	60673	17.55	47.79														

Figure 23. GTFPARS Output - Page 2 of 2.

Figure 24 on page 52 shows a condition code analysis in which '% CC 1' is Device Busy and Control Unit Busy expressed as a percentage of total SIOs.

DEV ADR	TOTAL SIO	DEVICE BUSY	CONTROL UNIT BUSY	% CC 1
250	1845	156	201	19.3
252	14516	60	296	2.5
253	366	15	32	12.8
254	1708	149	185	19.5
255	9547	698	858	16.3
256	3283	225	271	15.1
257	2119	208	246	21.4
370	8667	115	45	1.8
372	3294	90	149	7.3
373	403	3	6	2.2
374	1689	66	86	9.0
375	17800	310	410	4.0
376	146	1	2	2.1

Figure 24. SIO Condition Code Analysis

After reviewing the data, the following observations can be made.

Observation #1

GTFPARS is reporting 'device busy' conditions. The busy bit in the CSW has different meanings when used by itself and in combination with other CSW bits. For example by itself it means 'device busy', with the Status Modifier it means 'Control Unit Busy' and with Control Unit End it means a pending Control Unit End interrupt has been cleared. The version of GTFPARS used to reduce the data shown here did not differentiate between the Busy bit by itself and the Busy bit with Control Unit End. A detailed analysis of the GTF I/O records showed there were no

Busy bit only conditions. So the 'device busy' conditions shown in this data can be ignored.

Observation #2

There appears to be a relatively high level of contention in control unit 25X. All devices except device 252 have a relatively high percentage of condition code 1's.

Observation #3

Control Unit 37X has a relatively low level of contention.

Conclusions/Recommendations

A review of the system configuration indicated the control units were configured as shown in Figure 25.

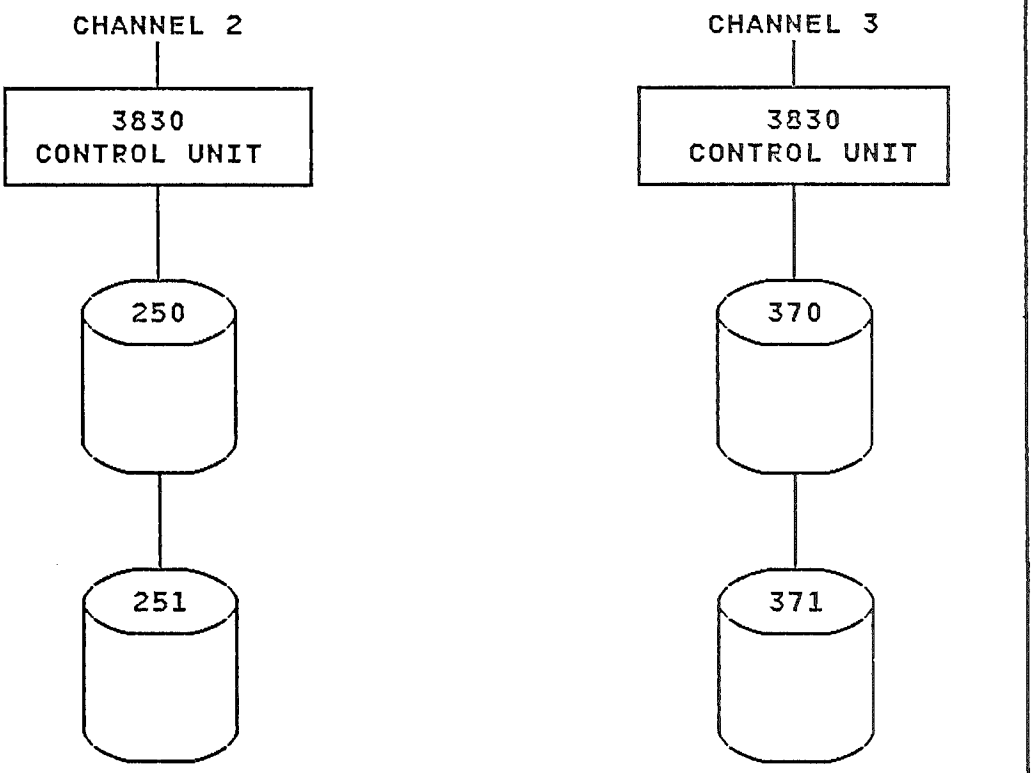


Figure 25. Problem Analysis Exercise - DASD Configuration

With only one channel path to a control unit, control unit busy conditions should not occur at the level shown for control unit 25X. However, further investigations of the configuration and the applications running when the data was collected, showed that:

1. All the drives on control units 25X and 37X were 3330 Model I (no release on a Format Write).

2. There was a high level of Format Write activity to device 252.

The Format Write activity was determined by doing a detailed analysis of the GTF data using the Print Dump Service Aid. An example of the sequence seen in the data is shown in Figure 26.

```
SIO  0252 ASCB 00FE50E8  CPU 0000  JOBN CICSDBLD  R/V CPA 002F6AD0 00FF6AD0  CAW 8000A9B0  DSID 00A4BF9C
          FLGS 00000000 1002       STAT 0C00      SK ADDR 00000000 08000D01  CC 0
          TIME    66504.079314
SIO  0250 ASCB 00011C00  CPU 0000  JOBN *MASTER*  R/V CPA 0006A828 0006A828  CAW 0000A978  DSID 00000000
          FLGS 00000010 8801       STAT 5000      SK ADDR 00000000 03000404  CC 1
          TIME    66504.108583
IO   0252 ASCB 00FE50E8  CPU 0000  JOBN CICSDBLD  OLD PSW 070C0000 00037064  TCB 00A42850  DSID 00A4BF9C
          CSW  802F6B00 0C000000   SNS  N/A       R/V CPA 002F6AD0 00FF6AD0  FLG 00001002  A2000252 00
          TIME    66504.111563
SIO  0250 ASCB 00011C00  CPU 0000  JOBN *MASTER*  R/V CPA 0006A828 0006A828  CAW 0000A978  DSID 00000000
          FLGS 00000010 8801       STAT 3000      SK ADDR 00000000 03000404  CC 1
          TIME    66504.112561
SIO  0252 ASCB 00FE50E8  CPU 0000  JOBN CICSDBLD  R/V CPA 001604E8 00FE24E8  CAW 8000A9B0  DSID 00A4BF9C
          FLGS 00000000 1002       STAT 3000      SK ADDR 00000000 08000D02  CC 0
          TIME    66504.113240
```

Figure 26. GTF Print Dump Output

Notice that the SIO to device 250 was rejected (CC 1) because of a Control Unit Busy condition. There are no other channel paths to this control unit, so we can assume a data transfer is not occurring. (If a data transfer were occurring, a Channel Busy would have been detected and the SIO would never have been issued.) Also note that three milliseconds after the SIO was attempted to device 250, an I/O Interrupt from device 252 was received. One millisecond later the pending Control Unit End interrupt was cleared by the second SIO attempt to device 250.

We can deduce from this data that device 252 and the control unit were busy but the channel was not. Because these are 3330 Model I devices, Format Write activity is probably the cause. This theory was substantiated when it was determined an application was creating a file on 252 during the time the data was collected.

This probably accounts for the low condition code 1 count for device 252 and the relatively high number of condition code 1's for the other devices on that control unit. It would appear a condition code 1 is being encountered when the other devices are attempting a SIO during the time 252 is erasing the track. Its not clear that anything other than migrating to 3330 Model 11 or 3350 devices can be done to reduce this delay.

When reviewing GTFPARS output, check the devices that violate the 25 to 40 millisecond service time rule against the SEEK DISTANCE SUMMARY. For example, device 253 shows an average service time of 41.988 milliseconds. The SEEK DISTANCE SUMMARY indicates an average seek distance of 114.22 Cylinders. The

high average seek distance may be the cause of the high service time.

To summarize, the approach to analyzing I/O problems illustrated here suggests the analyst begins by examining response time. This method suggests the following steps.

1. Start with the RMF Device Activity Report. Scan the data and make note of all the devices that appear to have excessive queue time, and service time.

2. For each device noted in step 1, calculate response time (queue time plus service time).

3. Order the devices by response time (highest first). Devices with the highest response time probably require attention first.

4. Additional measurements may be needed to further isolate the cause of the problem. Where control unit and/or arm contention is suspected, GTFPARS may be appropriate.

SERVICE LEVEL REPORTER

The Service Level Reporter Program Product (SLR Program Product 5740-DC3) can be a useful tool for analyzing I/O data. This program product provides the capability of manipulating SMF, RMF and IMS log tape data and then producing user defined reports.

Figure 27 on page 57 shows a sample report in which response time and its various elements were calculated from RMF data from multiple systems. The advantage in combining multiple systems is that one can see the affect one system has on the other. For example, refer to device 23A in Figure 27 on page 57. Note that 'SY 5' has the highest SIO rate and as such appears to be dominating the device. This is indicated by the fact that 'SY 5' is encountering a relatively low average Reserve Delay time while the other systems are encountering a substantially higher average Reserve Delay time. The same appears to be true for 23B from 'SY 4'.

RMF MEASURED DEVICE I/O

CUADDR	SYSID	IORATE IO/SEC	AVGRESPTIME MILLISEC	AVGSERVTIME MILLISEC	AVGQUETIME MILLISEC	AVGCUDELAY MILLISEC	AVGRESVDELAY MILLISEC
2D7	SY 6	0.000	---	---	---	---	---
2D8	SY 1	0.023	26.8	17.9	8.9	3.0	0.0
	SY 3	0.060	71.0	64.7	6.3	0.8	5.5
	SY 4	0.102	53.8	43.6	10.2	3.4	4.1
	SY 5	0.000	---	---	---	---	---
	SY 6	0.021	40.0	30.0	10.0	3.3	3.3
2D9	SY 1	4.327	49.5	26.6	22.9	0.6	10.0
	SY 3	0.027	25.8	23.2	2.6	2.6	0.0
	SY 4	1.289	37.2	17.8	19.4	1.7	12.8
	SY 5	0.230	39.5	18.4	21.1	1.8	18.7
	SY 6	0.967	36.4	18.6	17.8	0.4	15.4
23A	SY 1	0.034	47.5	28.9	18.6	2.1	16.5
	SY 3	0.357	50.8	29.8	21.0	0.2	20.0
	SY 4	0.331	62.0	33.8	28.2	6.5	18.7
	SY 5	3.716	54.8	45.0	9.8	2.8	3.0
	SY 6	0.262	58.3	35.8	22.5	0.8	19.1
23B	SY 1	0.031	69.2	46.9	22.3	6.7	13.4
	SY 3	0.006	62.5	50.0	12.5	0.0	12.5
	SY 4	2.098	78.6	69.5	9.0	5.3	2.2
	SY 5	0.355	50.5	32.3	18.2	2.0	13.9
	SY 6	0.081	60.8	37.7	23.1	1.7	15.4
23C	SY 1	0.010	42.9	42.9	0.0	0.0	0.0
	SY 3	0.000	---	---	---	---	---
	SY 4	0.048	42.2	37.8	4.4	2.9	1.5
	SY 5	0.043	24.2	22.6	1.6	0.0	0.0
	SY 6	0.061	28.4	19.3	9.1	2.3	0.0
23D	SY 1	0.056	49.3	41.9	7.4	1.2	2.5
	SY 3	0.524	35.8	31.9	3.8	0.8	2.8
	SY 4	0.091	34.2	30.4	3.8	1.5	0.0

Figure 27. Service Level Reporter Sample Output

A tool such as the SLR, can be a valuable productivity aid in that it can assist the system analyst in very quickly identifying trouble spots in an I/O configuration.

This sample report was produced by Service Level Reporter, Release 1 Modification Level 2, installed at the Washington Systems Center.

THIS PAGE INTENTIONALLY LEFT BLANK

OVER INITIATION

Over initiation has a negative effect on performance. The impact comes from the CPU cycles required to manage additional address spaces, greater demand on the spool and work packs, and a greater demand for real storage. The result is higher utilizations for spool, paging and work packs along with higher utilization for the channels that service them.

The solution to this problem is not to over initiate. However this is not always possible. Take, for example, the RJE system in which each RJE user insists on his own initiator. Therefore, the next best approach is to reduce the impact of over initiation.

As it turns out, generally the cost, in CPU cycles, of managing the additional address spaces is not as great as one would think it to be. A recent study on a 168 showed the cost of over initiating by six initiators was about 3%.

The real cost in over initiating comes from the result of the greater demands placed on the spool and work packs and the paging subsystem. If this could be controlled, the impact of managing the additional address spaces is acceptable.

The key to the over initiation problem is controlling the MPL level. This can be done by tuning the paging configuration. Even though extra initiators have been started, a tuned paging configuration, as described earlier, will keep the MPL at a proper level, thus allowing only the appropriate number of address spaces in storage. The paging, spool and work packs should then be defined in sufficient quantity to handle that level of work, using the previously stated ROTs. In this situation, exchange swapping should be monitored. If the number of exchange swaps appears excessive, it may be necessary to increase the ISV value in the IPS.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial attempts at improving performance in environments where the CPU is the constraining resource should be directed towards two specific areas: I/O blocksizes for sequential data sets and supervisor services. That's not to say these are the only places to look for CPU problems. It is, however, our experience that these are the areas most likely to be the cause. They also have the greatest potential for reducing CPU utilization.

SEQUENTIAL DATA SET BLOCKSIZES

In any discussion of blocksizes it must be clearly understood that we are talking about sequential data sets, not random or directly accessed data. With sequential data we know that the next block of data we want follows the one just processed. With random or directly accessed data this is not necessarily true. The data is generally not accessed in sequence. Therefore, blocking randomly accessed data can actually reduce performance by increasing storage requirements for larger blocksizes containing data that may not be referenced and the additional channel time required to transfer the larger records.

An evaluation of an installation's sequential data set blocksizes is unquestionably the first thing to do when trying to reduce CPU utilization. No other activity has anywhere near the potential for saving CPU cycles. To illustrate that point the following data shows the effect on a file maintenance program when going from a 200 byte to 6K blocksize. Four QSAM files were involved for a total of about 100,000 EXCPs at the 200 byte blocksize (BUFNO = 5 in all cases).

BLOCKSIZE	200	1000	2000	4000	6000
EXCPs	105,000	21,000	10,500	5,250	3,500
SIOs	51,319	10,468	5,277	2,728	1,800
Elapsed Seconds	600	150	103	71	56
CPU Seconds	73.6	21.6	14.3	10.6	9.3
Channel Seconds	95.2	46.9	38.4	36.5	33.3

Figure 28. Effect Of Reducing I/O Blocksizes

Note that in this example increasing the blocksize from 200 bytes to 6K reduced CPU utilization by more than 87% . In addition elapsed time dropped by more than 90% and channel time dropped 65% .

The preparation and execution of a plan to evaluate and make changes to an installation's data sets will take a certain amount of time and effort to complete. But the potential for reducing CPU cycles is substantial. There are, however, some relatively easy things that can be done initially. VIO for temporary data sets, whose blocksizes are less than 2K, can usually provide savings in CPU cycles. Also an examination of procedure libraries often turn up data sets whose blocksizes are less than optimum.

As far as recommended blocksizes are concerned, generally no blocksize either tape or disk, should be less than 4K. A more practical blocksize for disk is 6K primarily because it provides compatibility between 3330 and 3350 type devices. However, full track blocking for program libraries is still recommended. For tapes, blocksizes of 8 or 12K are generally quite reasonable. For some applications, when 6250 BPI tapes are available, 32K blocksizes may be appropriate.

A common concern to increasing blocksizes is that it will increase the paging rate to the point where it will adversely effect performance. Assuming your paging configuration is properly tuned, (this subject is dealt with previously), the evidence clearly indicates increased blocksizes generally have a positive effect on system performance. For example, one test case in which multiple copies of a high I/O job were run and blocksizes increased from 2K to 12K (full track), showed that although paging increased 43% (24.6 pages/sec to 35.2 pages/sec), elapsed time dropped by 9% , and CPU seconds dropped almost 12% .

A recent survey of 281,029 user data sets representing 52 MVS installations showed the following:

- 65 to 70% of the user data sets were sequential (SAM).
- Sequential Blocksize distribution
  - 85% of sequential data blocks are 4K or less.
  - 70% are 2K or less
  - 40% are 500 Bytes or less

It would appear from this study that there are quite a number of data sets that could be reblocked for better performance.

The best source of information about data set blocksizes comes from the SMF. If there is no readily available SMF reduction program, it may be necessary to write a simple program to extract the required information. Generally the most useful information consists of:

- Data set name
- Volume serial number
- Device type
- Blocksize
- Number of EXCPs

There is a certain amount of overhead associated with collecting SMF data. Unless this information is required for accounting systems, you may not want to collect these records on an on-going basis. In these situations, a period of time should be established for gathering the required data. Make certain this time period is long enough to collect data for major installation applications. In some cases several time periods may have to be defined so that information relative to monthly applications, for example, is collected.

Once the data is collected and reduced, you may find about 10% of your data sets worthy of some remedial action. Obviously priority should be given to those data sets with the smallest blocksize and the largest number of EXCPs.

As was stated earlier, the first place to look for CPU cycle savings is data set blocksizes. No other tuning activity has anywhere near the potential for reducing total CPU cycles.

SUPERVISOR SERVICES

Unnecessary use of supervisor services can waste processor cycles. Although the impact is generally not as great as poor blocksizes, it may be worth some investigative effort. Here you are generally working with application programs. The objective is to determine what, if any, applications are using an inordinate number of services.

Again GTFPARS can provide useful information in helping to determine the impact of supervisor services. Data on the number and kind of SVCs issued along with SVC timings is provided. Input to GTFPARS is, of course, GTF trace data.

Because of the system usage associated with GTF tracing, one must select the appropriate time to activate the trace. This

is probably the most critical step in the analysis. The trace should be activated during those periods when poor performance has been observed. To minimize GTF tracing, only the required events should be collected. In general, the trace should be active long enough to collect one reel of tape (about ten to fifteen minutes on a 3032 or 168). Several periods may have to be traced to get a reasonable profile of the system.

Information about 'SVC RATE' and 'EXCP RATE' can give some initial insight into the system profile. For example on a 3031 (or 158) on e would not expect to see SVC rates of over 500 per second or EXCP rates of over 100 per second. Rates near these values would indicate further investigation is appropriate. Comparable values for the 3032 (or 168) are SVC rates of 1200 per second and EXCP rates of 300 per second. As of now, there isn't enough detailed information on the 3033 to make a general statement concerning SVC and EXCP rates. However, if one were to venture a guess, numbers like 2000 SVCs per second and 500 EXCPs per second might appear.

Certainly there are a variety of conditions that can exist during the analysis of GTFPARS output too numerous to examine in detail here. What the preceding paragraphs have attempted to do is describe one approach to the collection and initial analysis of GTFPARS output.

MISCELLANEOUS COMMENTS

Inappropriate use of SMF and RMF parameters can cause excessive CPU utilization. As far as SMF is concerned, its difficult to be precise because the overhead is a function of the kind and type of records collected. As a general rule, each installation should examine the kind of data it needs and ensure only required records are collected. For example, few users require the Type 40 (Dynamic Allocation) record and yet referring to the TSO/Batch data we find that 25% of all the SMF Records are Type 40. Eliminating them could reduce SMF's requirements. The solution in this case would be to write an IEFU83 EXIT and discard the unwanted records. A sample IEFU83 EXIT can be found in SYS1.SAMPLIB in both the IPO and non-IPO Systems. With MVS/SE Release 2,the impact of SMF on the system has been reduced. Also, additional controls have been provided so that the collection and writing of unwanted records can be eliminated through SMF parameters. For more information on SMF and MVS/SE Release 2 refer to Washington Systems Center Technical Bulletins 'MVS/SE Release 2 Product Presentation' (GG22-9029) and 'MVS/SE Release 2 Planning Information' (GG22-9028).

When using RMF for long term measurements, as described in the preceding chapter , the impact on storage and CPU is generally low. However, because of the impact of the TRACE function,

users should be extremely careful in selecting the proper options for short term measurements.

THIS PAGE INTENTIONALLY LEFT BLANK

One characteristic of a system with a storage contention problem is the inability to fully utilize the processor (CPU). In some cases it may not be possible to get CPU utilization above 60% .

The basic solution to a storage constrained system is more real storage. If you have a 4 megabyte IMS system and only 3 megabytes of storage to run it in, no amount of parameter adjusting, SRM modifications or system zapping will make it run well. What will make it run well is 4 megabytes of storage, assuming you've tuned buffers for system components such as TCAM, VTAM, VSAM, IMS, etc.

However, in many cases, where multiple subsystems are involved, you can tune the system to run optimally with the storage it has. The specifics of this process were discussed in the chapter 'Paging Configuration Tuning.' Again, tuning a storage short system will generally result in a CPU that is not fully utilized.

When tuning a system with insufficient real storage, you are generally dealing with a series of trade-offs. For example you may need to reduce the number of batch initiators in order to achieve reasonable TSO or IMS response. Therefore, it is extremely important that the issue of workload priorities as discussed in the chapter 'Performance Management' has been addressed and resolved.

A report entitled 'Memory Can Increase Your Capacity' (Form # GG22-9053) describes the relationship between real storage and system capacity. Included are measurements from various user environments showing the effect on performance of adding more real storage. One example of two MVS users increasing their 5 megabyte 168 to 6 megabytes showed the following results:

- Paging decreased 43% (From 43.3 to 24.8 pages/second)
- 25% increase in batch throughput (From 53 to 66.5 jobs/hour)
- 33% reduction in average TSO response (From 4.4 to 2.9 seconds)
- 20% reduction in average IMS response (From 5 to 4 seconds)

For more specifics on how additional storage can effect system capacity, refer to the above mentioned technical bulletin.

THIS PAGE INTENTIONALLY LEFT BLANK

The preceding pages have highlighted some of the MVS tuning experiences of the D P Washington Systems Center. To net it out, there are two points that can be concluded from that experience. First, the reason an installation finds itself in the midst of a performance crisis is generally because its performance management system is inadequate. Many performance problems can be avoided if objectives are set and an effective monitoring system is in place.

Second, I/O contention of some sort is usually the cause of MVS performance problems. Approximately three quarters of the problems the DP Washington Systems Center has been involved in, were caused by I/O bottlenecks. A good rule to follow is to examine the I/O subsystems first.

As was stated in the Introduction, the rules of thumb described in this document relate to 'reasonable' performance. However, what's reasonable in one situation may not be reasonable in another. What's needed is a methodology to determine 'reasonableness' for a given subsystem, installation, etc.

The 'DASD Configuration and Sharing Considerations' technical bulletin (see References) describes such a methodology. Basically, the methodology suggests a data collection phase, using a sampling technique, and then plotting the data to determine if a correlation exists between, for example, response time and channel utilization. Having found a correlation, it could then be determined at what point channel utilization causes the response time to exceed the performance objective.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. REFERENCES

TECHNICAL BULLETINS

- DASD Configuration and Sharing Considerations (GG22-9052)
- 303X Channel Configuration Guidelines (GG22-9020)
- Memory Can Increase Your Capacity (GG22-9053)
- MVS/System Extensions Release 2.0 Planning Information (GG22-9028)
- MVS/System Extensions Release 2.0 Product Presentation (GG22-9029)

SRLS

- Initialization and Tuning Guide (GC28-0681)
- Performance Notebook (GC28-0886)
- MVS Data Management Services Guide (GC26-3875)

THIS PAGE INTENTIONALLY LEFT BLANK

The following describes how the SAM-E (Sequential Access Method - Extended) program product, 5740-AM3, provides benefits over the previous SAM logic. It was written by Paul Dorn of the Washington Systems Center and originally appeared as a WSC Flash.

SAM-E uses EXCPVR to process most direct access and VIO operations for BPAM, BSAM and QSAM requests. This technique allows SAM-E to minimize the number of system services required to process the user requests.

SEARCH DIRECT

SAM-E forces the use of "Search Direct" logic for input datasets, whenever possible. Search direct means that the channel program searches for the first record to read. To determine the address of the record to read, every search direct channel program ends in a read of the address of the next record in the dataset ("Read Count" CCW). Without search direct, the channel program searches on the record previously read (known as "search previous") and then reads the next record.

A non-keyed dataset on disk has the following format:

R1 CT	GAP	R1 DATA	GAP	R2 CT	GAP	R2 DATA
21	82	4000	82	21	82	4000

Normally the data field of each record is much larger than the count or gap portions. For a 3350, the count field uses 21 bytes and each gap is equivalent to 82 bytes. Assume for the remain-

[6] This is a simplified version of the channel program and does not include all the required CCWs. For example the SEEK and SET SECTOR commands are not shown. For more information on the actual channel programs, see OS/VS2 MVS Sequential Access Method - Extended (SAM-E) Logic, LY26-3885.

der of this discussion that the data portion is 4000 bytes. Also, a 3350 can process a byte of data every 835 nanoseconds.

Without search direct, the following channel program is required to read record number two:[6]

```
SEARCH ID EQ rcd1
TIC *-8
READ COUNT rcd2
READ DATA rcd2
```

The channel and control unit are tied up from the time the device "finds" record 1 to the time it is able to read record 2. For our example, this is the time required to pass over records 1 and 2, and their associated counts and gaps. In our example this is 8288 bytes (21 + 82 + 4000 + 82 + 21 + 82 + 4000), which requires 6.9 milliseconds.

With search direct, the following channel program is required to read record number two:[6]

```
SEARCH ID EQ rcd2
TIC *-8
READ DATA rcd2
READ COUNT rcd3
```

The channel and control unit are tied up from the time the device "finds" record 2 to the time it is able to read the count field of record 3. For our example, this is the time required to pass over record 2 and the gap and count fields associated with records 2 and 3. In our example this is 4206 bytes (21 + 82 + 4000 + 82 + 21), which requires 3.5 milliseconds. It is obvious that this is a major savings.

Naturally, this is not all the time required to read a record. However, it does account for most of the time the control unit and channel are busy. From this discussion it is obvious why "search direct" can significantly improve performance.

VIRTUAL CHAINED SCHEDULING

Prior to SAM-E, virtual chained scheduling was used to increase the number of buffers processed per I/O request. Virtual

chained scheduling is best explained with an example. Assume an output dataset is processed using QSAM, and five buffers are defined (the default). For simplicity, assume the dataset is unblocked; that is, each PUT results in a WRITE of a new block. The following is a typical scenario for this processing:

1. The application issues a PUT. QSAM converts this into a WRITE of record 1..

2. The application issues a second PUT. Since the first I/O is still not complete, SAM does not start the second WRITE.

3. The application issues a third PUT. Since the first I/O is still not complete, SAM does not start the third WRITE.

4. The first I/O completes. SAM now starts a write of records two and three.

5. This process continues with each I/O processing two buffers.

Thus, in this example chained scheduling reduces the number of I/O requests by fifty percent. The actual improvement is dependent upon the following factors:

- If the records are blocked, or the program does significant processing on each record, it is very likely that many of the I/O requests process only one buffer.

- If the system load is heavy, the program may not be able to generate/require records often enough to schedule multiple buffers per I/O.

In any case, the **best** virtual chained scheduling can do is an average of BUFNO/2 buffers per I/O (assuming the default of 5 buffers, the best is 2.5 buffers per I/O). Typically, the average number of buffers per I/O is much less.

Unfortunately, virtual chained scheduling has an additional problem: it cannot be used if the "search direct" option is selected. Therefore, to read the 2.5 records, the channel and control unit must process 3.5 records (includes the record found by the "search previous"). For more information on search direct and virtual chained scheduling, see the manual OS/VS2 MVS Data Management Services Guide, GC26-3875.

SAM-E, on the other hand, insures that each QSAM I/O request processes multiple buffers. With SAM-E, the previous scenario would become:

1. The application issues a PUT. SAM-E places the request on its internal queue.

2. The application issues three more PUTs. SAM-E also places the requests on its internal queue.

3. The application issues a fifth PUT. Since all the buffers are now ready for processing, SAM-E generates an I/O request which writes out all five buffers.[7]

4. This process continues with each I/O processing five buffers.

Thus SAM-E guarantees the maximum number of buffers are processed per I/O request. It is not affected by the design of the application or the system load. It also works in conjunction with the "search direct" option previously described.

Naturally, this technique only works for QSAM. For BSAM, SAM-E uses the standard chained scheduling technique (although search direct is still used in this case).

Processing multiple buffers per I/O is probably the most important performance improvement in SAM-E. Following is a comparison of the time spent doing two I/O's to read four buffers (prior to SAM-E), and one I/O to read four buffers (with SAM-E). The 3350 values previously described are also used in this example. We will assume no contention for the control unit or channel (a situation which seldom occurs in the real world). Also, we assume the DASD arm is not "stolen" between I/O requests. Note that there are additional CCWs in the following channel programs which are not shown. Normally they add less than 1 millisecond of channel time.

The following would be the average time required to issue the two non SAM-E requests (in milliseconds):

---

[7] SAM-E will schedule an I/O request when its internal workarea is full. Since approximately 13 buffers can be processed using this workarea, SAM-E almost never issues an I/O unless all the buffers are full. SAM-E will also schedule an I/O when it has approximately 64K bytes of data to transfer.

Function	Channel	Ctl Unit	Device	Comments
SEEK	0	0	0	no arm move
SET SECT	.4	.4	8.4	1/2 rotation
SEARCH 1	3.4	3.4	3.4	incl pass over 1
READ 2	3.5	3.5	3.5	gap, ct, data
READ 3	3.5	3.5	3.5	gap, ct, data
SEEK	0	0	0	no arm move
SET SECT	.4	.4	8.4	1/2 rotation
SEARCH 3	3.4	3.4	3.4	incl pass over 3
READ 4	3.5	3.5	3.5	gap, ct, data
READ 5	3.5	3.5	3.5	gap, ct, data
TOTAL	21.6	21.6	37.6	

The following would be the average time required to issue the one SAM-E request for 4 records (in milliseconds):

Function	Channel	Ctl Unit	Device	Comments
SEEK	0	0	0	no arm move
SET SECT	.4	.4	8.4	1/2 rotation
SEARCH 2	.2	.2	.2	find record
READ 2	3.4	3.4	3.4	gap, ct, data
READ 3	3.5	3.5	3.5	gap, ct, data
READ 4	3.5	3.5	3.5	gap, ct, data
READ 5	3.5	3.5	3.5	gap, ct, data
RD CT 6	.1	.1	.1	for next read
TOTAL	14.6	14.6	22.6	

From this example it is obvious that reducing the number of I/O requests can significantly reduce the I/O subsystem load. This becomes even more important as the load on the subsystem grows. If contention exists for the control unit, the device will have problems reconnecting to the control unit, and require additional revolutions before transferring the data. For the non SAM-E case, there are two reconnects (one in each channel program when the set sector is complete). In the SAM-E case, only one reconnect is required to process four records. Similarly, arm "stealing" can occur twice in the non SAM-E case, and only once in the SAM-E environment (that is, each channel program may require the arm to be moved to locate the desired cylinder).

AUTOMATIC BLOCKING

For QSAM files, SAM-E can be considered an "automatic blocking" function. For example, a dataset with DCB characteristics of:

```
DCB=(RECFM=F,LRECL=80)
```

is handled by SAM-E with each I/O processing 5 blocks. This is equivalent to having a non SAM-E processed dataset with the DCB characteristics of:

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=400)
```

Many customers feel all their datasets are blocked, and therefore the SAM-E "automatic blocking" will not improve their system performance. In fact, almost all installations have a large number of files that are poorly blocked for sequential access:

- Utility program work files
- Intermediate files used by applications
- Database files used for direct access (and therefore want small blocksizes) which are backed-up for recovery purposes using QSAM reads.

With additional investigation most customers can probably find other examples where SAM-E "automatic blocking" can improve system performance.

OBSERVATIONS

The technique SAM-E uses to process multiple QSAM buffers per I/O request can cause some programs to "gallop." That is, the program executes, then goes into a wait state, then executes, etc. When run without SAM-E, the program is able to overlap its CPU and I/O processing. This is caused by SAM-E not overlapping QSAM I/O with the execution of the user's program (assuming a single TCB structure). However, in a most systems the galloping should not increase the average elapsed execution time of the program since the normal multiprogramming function of MVS will allow the CPU to be used while the one program is waiting.

If this galloping causes a problem, requesting more than 13 buffers (DCB=BUFNO=xx) or more than 64K bytes of data (e.g., BLKSIZE=12960,BUFNO=6) will cause I/O and CPU overlap.

APPENDIX C. SOME THOUGHTS ON IPS DESIGN

What follows are this author's views on IPS design. They are based on personal experience and may or may not be consistent with others whose experiences are different. They are presented here for your consideration and evaluation.

If you recall, the Introduction stated that the intent of this document was to address those areas of system performance that tend to be common problems. If you scan the table of contents, you will notice that there isn't much that talks to IPS design. That should be a clue to the reader that, at least from this author's perspective, IPS design is not a major problem area. Very seldom do we find a bad IPS causing major performance problems. When we do find IPS problems, they tend to be gross rather than subtle design errors.

However, we do find installations spending a great time on IPS design (much more time, in this writers opinion, than it deserves). One reason might be that the SRM provides us with a variety of options which we, as technicians, feel compelled to use because they are there. However, the intent of these options are to address specific predefined needs. If the need is not there, the options should not be used. For example, one practice seen quite often is an attempt to discriminate workloads at such a fine level of detail that the effects cannot be measured. If the effects of your specifications cannot be measured or if the difference is meaningless (who cares if a batch job with a 24 hour turnaround time requirement gets done 1 hour faster), your efforts should probably be directed in more productive areas.

What follows is a description of an approach that attempts to define some realistic goals for IPS design. Again, the intent is not to describe the details of defining an IPS, but rather to provide some 'food for thought' on how one could approach IPS design.

THE APPROACH

The SRM provides controls for: resource utilization data for non-swappable address spaces, dispatching priorities, and swapping decisions. It is the IPS parameters that control the swapping decisions that are being addressed here. It is assumed that the reader understands the value of performance groups for the non-swappable subsystems and has applied some logic to assigning dispatching priorities to the various system workloads.

In most environments, swapping takes place for two kinds of workloads: TSO and batch. (The other subsystems such as IMS, CICS, JES, TCAM, VTAM, etc. are generally non-swappable and therefore the SRM makes no swapping decisions for them.) Within the batch and TSO workloads there are subsets. For TSO, there are generally trivial, medium, and long transactions. Batch may have several levels of turnaround requirements such as 30 minute, 2 hour, 8 hour, and 24 hours. Their very definition implies a priority which might be as follows:

1. TSO Trivial
2. TSO Medium
3. 30 Minute Batch
4. TSO Long
5. 2 Hour Batch
6. 8 Hour Batch
7. 24 Hour Batch

Figure 29. Sample Workload Definitions

Our task, therefore, is to ensure that under varying system loads, the SRM is given the right instructions for making swap decisions. In particular, we are concerned about TSO trivial and medium transactions because of their effect on user productivity and their visibility. With TSO our concern is in the area of seconds. On the other hand, the batch is generally less critical. Here we are dealing in terms of minutes and in some cases hours. This distinction is key to approach described here.

The TMPL (Target Multiprogramming Level)[8] is critical to the swap decision because it is the factor that determines the number of address spaces in real storage at any given instant in time. The first step is to ensure the system TMPL is correct under varying system loads. That process was discussed in detail in the section of this document entitled 'Paging Configuration Tuning'. You may want to review that material before proceding here since the remainder of this discussion assumes that those recommendations have been implemented.

[8] In this discussion, the term TMPL will refer to the Target MPL calculated by the SRM. The term MPL will refer to the value specified by the user.

Having ensured the system TMPL is correct, our job now is to distribute the system TMPL among the contending workloads. For this discussion we will use the workloads described above. There are seven distinct workloads identified. We now need to group these workloads into controllable units so that the SRM knows how to discriminate between them. One such grouping might be as follows.

1. TSO Trivial and Medium Transactions.
2. 30 Minute Batch
3. TSO Long Transactions and 2 Hour Batch
4. 8 Hour Batch
5. 24 Hour Batch.

Figure 30. Sample Workload Groupings

Admittedly there are other ways to group the workloads. Our example shows one possibility.

Because we want to control each of these workloads individually, a domain should be defined for each group (Total of 5 domains). You may want to define multiple objectives to differentiate, for example, between medium and trivial TSO transactions.

The next step is to tell the SRM how to discriminate between the workloads. I suggest using Weights. We know from experience that in order to get measurable discrimination between workloads using Weights, the Weight values should be separated by at least 50. Our definition might look like this.

Domain Definition	Weight
Domain 1 - TSO Triv/Med	250
Domain 2 - 30 Min. Batch	200
Domain 3 - TSO Long/ 2 Hr. Batch	150
Domain 4 - 8 Hr. Batch	100
Domain 5 - 24 Hr. Batch	50

Figure 31. IPS Weight Defination

MPL specifications for the domains is the next step. Using weights generally favors TSO because the TMPL calculation for each domain considers 'Ready User Average' and as a group, there are usually more ready TSO users. In general, we find

that 20% of the logged on users want to be in storage at any given instant. Assuming a maximum of 60 logged on users, we would probably want our maximum MPL for Domain 1 set to 12. This would be an initial setting and should be verified with subsequent measurements. The minimum MPL for Domain 1 should probably be set to 6 (10% of maximum logged on users). This should ensure service to users logging on when the number of TSO users in the system is small.

In establishing initial min and max MPL values, keep in mind that the values for Domain 1 are the most sensitive. That is because TSO is response oriented and gross errors in the specifications will be noticed by the users very quickly. A minimum MPL that is too low will cause the system to react slowly to changing loads. (Remember the TMPL can change only every twenty seconds, and then only by a value of one.) A maximum MPL that is too high can cause TSO to dominate the system.

The MPL settings for the remaining domains will depend upon the average number of jobs per measurement period in each group. To establish the initial min and max MPL settings for the batch domains current RMF data should be used. Working set size for the average transaction within a domain along with workload priority should provide some insight into what a reasonable MPL should be. However, one should always try to give the SRM latitude for making swap decisions. For example, theoretically one could assign an MPL for Domain 5 as (0,255). This gives the SRM maximum flexibility with respect to Domain 5. You may want to temper that somewhat because it would be possible under heavy loads that Domain 5 may get a TMPL of 0 for some time. (1,20) may be a more appropriate starting place. Errors in batch specifications are less critical and can be identified and modified before it effects the user community.

To simplify the process, these initial settings should be made to the performance objectives provided in the IPS supplied with the system. There is no need at this point in time to spend time trying to adjust service rates and workload levels. Those adjustments can come when and if the subsequent measurement data indicates a modification is required.

Having gone through our exercise, the initial specifications may now look something like the following.

Domain Definition	Weight	MPL
Domain 1 - TSO Triv/Med	250	(3,12)
Domain 2 - 30 Min. Batch	200	(2,20)
Domain 3 - TSO Long/ 2 Hr. Batch	150	(3,20)
Domain 4 - 8 Hr. Batch	100	(2,20)
Domain 5 - 24 Hr. Batch	50	(1,20)

Figure 32. IPS Weight Defination

The important thing to remember at this point is that the values specified are initial. Measurements must now be made to validate your initial estimates.

To recap some general guidelines for establishing initial MPL settings:

1. Use the 20% Rule for initial TSO domain maximum MPL.

2. Estimate minimum and maximum MPLs for the batch domains. For these initial settings try to give the SRM as much latitude as possible for adjusting to varying system loads. Keep the minimum MPL to some small non-zero number (like 1 or 2) under heavy system loads you truly do not want that domain to get service. Maximum MPL should be set initially to a value that exceeds a level that would normally be reached. The weight values should prevent a given domain from dominating the system.

3. Run the system and measure these initial setting. Track 'Ready User Average', MPL, and 'Target MPL'. In a responsive system, the 'Ready User Average' should approximate the 'Target MPL'.

   A Background Monitor II Session is probably a good idea particularly while the system is being brought up the first time. It will allow you to pick up any gross problems with the IPS so that adjustments can be made quickly.

   This measurement step is the most important step in the process. Under no conditions should any changes be made to an IPS without 'before' and 'after' measurements. One must verify that the intended change has in fact occurred and no other element has been adversely affected.

4. Make any adjustments the measurements suggest. Then continue to monitor periodically.

One other consideration is to ensure exchange swapping is eliminated for TSO trivial and medium transactions and is kept to minimum for batch. This is best controlled by the ISV parameter. Assuming your performance groups are defined for the various workloads, then the average service per transaction type can be determined from the RMF Workload Report. This could be used as the initial ISV. If your measurements indicate exchange swapping is excessive, then the ISV should be increased.

If you have followed the recommended approach, you should now have a relatively effective IPS. From this point on any effort applied to further IPS tuning will very quickly reach the point

of diminishing returns. The amount of 'useful' payback compared to the effort expended is usually low.

SUMMARY

The underlying premise of this discussion has been that the best IPS is a simple IPS. Complex IPSs take a great deal of time to develop and very seldom, if every , do they provide effective improvements. The approach described here suggests the following steps.

1. The first rule is 'Keep It Simple'. Make the initial changes to the default IPS. Use the various SRM options when and only when you have a specific predefined need (i.e., trying to solve a specific problem).

2. Define Domains for each unit of unique controllable work within the system. Use the Weights to discriminate between workloads.

3. Use the ISV value to control exchange swapping.

4. Measure all changes made to the IPS. Don't assume that what you intended to do has in fact happened.

With MVS/SE Release 2 substantial enhancements have been made to the SRM. These enhancements offer significant usability improvements particularly in the area of data collection. For example, the resources used by a specific group of TSO users can be collected. This type of information can be extremely useful in understanding resource utilization, and therefore, provides the potential for improving performance management.

The basic premise of this discussion is that one can get maximum system throughput while maintaining acceptable online/interactive response without spending inordinate amounts of time trying to define elaborate IPSs. To accomplish this it is assumed the installation has categorized and prioritized its workloads. It also assumes there are adequate system resources to support the installation workloads. For example, if there is insufficient real storage to support a required MPL, no amount of IPS manipulation can solve the problem.

APPENDIX D. PRESENTATION FOILS

The set of foils that follow is essentially an outline of the material presented in this document. Included in the presentation summary is a list of the Rules of Thumb. The document itself should be considered the presentation script.

IBM
DP DIVISION
WASHINGTON
SYSTEMS
CENTER

AN MVS TUNING PERSPECTIVE

AN MVS TUNING PERSPECTIVE

PRESENTATION INTENT

- SUGGEST WAYS TO MANAGE PERFORMANCE
- SUGGEST TUNING APPROACH

AN MVS TUNING PERSPECTIVE

RULES OF THUMB

- BASED ON EXPERIENCE
- GUIDELINES, NOT ABSOLUTE LAWS
- DESCRIBE REASONABLE PERFORMANCE

AN MVS TUNING PERSPECTIVE

KINDS OF PERFORMANCE PROBLEMS

- PERFORMANCE MANAGEMENT
- CONTENTION
  - I/O
  - STORAGE
- SATURATED CPU

AN MVS TUNING PERSPECTIVE
PERFORMANCE MANAGEMENT

INEFFECTIVE PERFORMANCE MANAGEMENT

- GENERALLY ENVIRONMENTS WHERE:
  - THERE ARE NO PERFORMANCE OBJECTIVES
  - WORKLOADS NOT PRIORITIZED
  - SYSTEM PERFORMANCE NOT MONITORED
- USUALLY THE UNDERLYING CAUSE OF A PERFORMANCE CRISIS

AN MVS TUNING PERSPECTIVE
PERFORMANCE MANAGEMENT

SOME ELEMENTS OF A GOOD PERFORMANCE MANAGEMENT SYSTEM

- DOCUMENTED PERFORMANCE OBJECTIVES BY WORKLOAD TYPE
  - DEFINE USER SERVICE
    - USER EXPECTATIONS
  - TUNING OBJECTIVES
  - SOME EXAMPLES:

WORKLOAD	OBJECTIVE
IMS	
SIMPLE TRANSACTION (5 OR LESS DL1 CALLS)	2 SEC RESPONSE
MEDIUM TRANSACTION (30 OR LESS DL1 CALLS)	5 SEC RESPONSE
COMPLEX TRANSACTION (200 OR LESS DL1 CALLS)	30 SEC RESPONSE
TSO	
TRIVIAL TRANSACTION (150 OR LESS SERVICE UNITS)	2 SEC RESPONSE
MEDIUM TRANSACTION (150 TO 2K SERVICE UNITS)	20 SEC RESPONSE
LONG TRANSACTION (GREATER THAN 2K SERVICE UNITS)	10 MIN RESPONSE

  - CONSIDERATIONS
    - PEAK LOAD VS AVERAGE

AN MVS TUNING PERSPECTIVE
PERFORMANCE MANAGEMENT

SOME ELEMENTS OF A GOOD PERFORMANCE MANAGEMENT SYSTEM (CONT'D)

- PRIORITIZED WORK
  - PEAK LOAD DECISIONS (SRM PARAMETERS)
  - BACK-UP CONSIDERATIONS

AN MVS TUNING PERSPECTIVE
PERFORMANCE MANAGEMENT

SOME ELEMENTS OF A GOOD PERFORMANCE MANAGEMENT SYSTEM (CONT'D)

- PERFORMANCE MONITORING
  - INDICATES IF OBJECTIVES ARE BEING MET
  - WORKLOAD TRENDS
    - ANTICIPATE PERFORMANCE PROBLEMS
    - CAPACITY PLANNING

AN MVS TUNING PERSPECTIVE
PERFORMANCE MANAGEMENT

SOME ELEMENTS OF A GOOD PERFORMANCE MANAGEMENT SYSTEM (CONT'D)

- KINDS OF MEASUREMENTS
  - LONG TERM
    - PERFORMANCE MONITORING
      - INCLUDES SYSTEM MEASUREMENTS AND REVIEW OF DATA
      - GENERAL SYSTEM PERFORMANCE DATA
      - WORKLOAD TRENDS
    - RUN CONTINUOUSLY
  - SHORT TERM
    - PROBLEM SOLVING
      - DATA COLLECTED FOR A SPECIFIC PROBLEM

AN MVS TUNING PERSPECTIVE
PERFORMANCE MANAGEMENT

SOME ELEMENTS OF A GOOD PERFORMANCE MANAGEMENT SYSTEM (CONT'D)

- LONG TERM MEASUREMENT RECOMMENDATIONS:

RMF PARAMETERS

CHAN
CPU
CYCLE(1000)
DEVICE(DASD)
INTERVAL(60M)
ENQ(SUMMARY)
PAGESP
PAGING
RECORD
NOREPORT
NOSTOP
NOTRACE
WKLD(PERIOD)
ARDJ/ASDJ (MAJOR SUBSYSTEMS AND COMPONENTS)

DATA TO TRACK

CPU UTILIZATION
CHANNEL UTILIZATION
DASD DEVICE UTILIZATION (CRITICAL DEVICES)
WORKLOADS
  RESPONSE TIME (TSO)
  TURNAROUND (BATCH)
  NUMBER OF TRANSACTIONS
  TRANSACTION RATE
PAGING ACTIVITY
  PAGE FAULTS
  DEMAND
  SWAP
  VIO
  PAGE DATA SETS
ADDRESS SPACE DATA
  CPU UTILIZATION
  I/O ACTIVITY
  STORAGE UTILIZATION
  PAGE FAULTS

AN MVS TUNING PERSPECTIVE
PERFORMANCE MANAGEMENT

SOME ELEMENTS OF A GOOD PERFORMANCE MANAGEMENT SYSTEM (CONT'D)

- SHORT TERM MEASUREMENT

RMF PARAMETERS

```
  CHAN
  CPU
  CYCLE(250), CYCLE(333) IF MVS/SE
  DEVICE(DASD)
  INTERVAL(30M)
  PAGESP
  PAGING
  RECORD
  REPORT
  NOSTOP
* TRACE
  WKLD(PERIOD)
* ENQ(DETAIL)
* ARDJ/ASDJ
```

* THESE PARAMETERS WOULD BE USED DEPENDING UPON THE KIND OF DATA NEEDED TO SOLVE A SPECIFIC PROBLEM

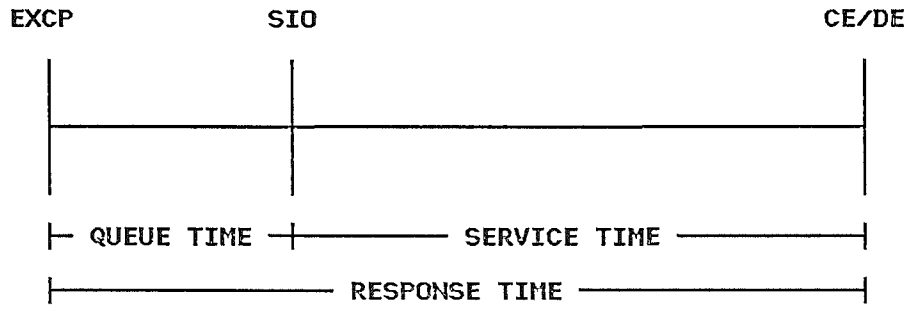
AN MVS TUNING PERSPECTIVE
I/O CONTENTION

- MANY MVS PERFORMANCE PROBLEMS ARE I/O RELATED
- FIRST PLACE TO LOOK FOR CAUSE OF PERFORMANCE PROBLEM
- PROBLEMS CENTER AROUND:
  - CHANNEL LOADING
  - CONTROL UNIT CONTENTION
  - ARM CONTENTION
- GENERALLY CAUSED BY:
  - DATA SET PLACEMENT
  - PAGE DATA SET CONFIGURATION
  - SHARED DASD

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

RESPONSE TIME

- KEY TO FINDING I/O PERFORMANCE PROBLEMS
- DEFINITIONS:



AN MVS TUNING PERSPECTIVE
I/O CONTENTION

RESPONSE TIME

- QUEUE TIME
  - RMF REPORTS AVG Q LNGTH FOR ALL DEVICES
    - INCLUDES QUEUING FOR:
      - DEVICE BUSY
      - CHANNEL BUSY
      - CONTROL UNIT BUSY
      - SHARED DASD/HEAD OF STRING BUSY
  - QUEUE TIME IS AVG Q LNGTH CONVERTED TO TIME
    - AVG Q LNGTH DIVIDED BY DEVICE ACTIVITY RATE EQUALS QUEUE TIME IN SECONDS
  - SHOULD BE KEPT TO MINIMUM ESPECIALLY FOR DEVICES SERVICING RESPONSE ORIENTED APPLICATIONS

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

RESPONSE TIME

- QUEUE TIME (CONT'D)
  - QUEUE TIME CAUSED BY CONTROL UNIT BUSY REPORTED BY RMF AS % CU DELAY
    - % CU DELAY CAN BE CONVERTED TO TIME, IN SECONDS, BY DIVIDING BY DEVICE ACTIVITY RATE
  - QUEUE TIME CAUSED BY SHARED DASD/HEAD OF STRING REPORTED BY RMF AS % RESV DELAY
    - % RESV DELAY CAN BE CONVERTED TO TIME, IN SECONDS, BY DIVIDING BY DEVICE ACTIVITY RATE
  - QUEUE TIME NOT CAUSED BY % CU DELAY OR % RESV DELAY CAN BE ASSUMED TO BE CAUSED BY DEVICE BUSY (UCB BUSY BIT ON)

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

RESPONSE TIME

- QUEUE TIME (CONT'D)
  - MOST SIGNIFICANT ELEMENTS OF QUEUE TIME ARE:
    - % RESV DELAY
    - DEVICE BUSY
  - CHANNEL BUSY AND CONTROL UNIT BUSY ARE MORE SIGNIFICANT TO SERVICE TIME

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

RESPONSE TIME

- SERVICE TIME
  - GENERALLY SHOULD BE 25 TO 40 MILLISECONDS FOR 3330/3350 NON PAGE DATA SET DEVICES
  - AFFECTED BY:
    - CONTROL UNIT BUSY
    - CHANNEL BUSY
    - LARGE VTOC AND/OR LIBRARY SEARCHES
    - ARM CONTENTION
  - TO CALCULATE SERVICE TIME, DIVIDE % DEVICE BUSY BY DEVICE ACTIVITY RATE
    - PROVIDED BY MVS/SE VERSION OF RMF AS AVG SERV TIME

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

- 3081 I/O QUEUING
  - QUEUES I/O REQUESTS IN HARDWARE FOR CHANNEL, CONTROL UNIT, AND/OR SHARED DASD/ HEAD OF STRING BUSY CONDITIONS
    - SIOF ONLY
    - ONE REQUEST PER DEVICE
  - OPERATION
    - IF CHANNEL OPERATIONAL, SIOF CONDITION CODE SET TO 0
    - IF BUSY CONDITION, REQUEST IS QUEUED
    - IF AFTER 10MS REQUEST IS STILL QUEUED, AN ATTEMPT TO START I/O IS MADE
      - IF UNSUCCESSFUL, DEFERRED INTERRUPT IS GENERATED AND REQUEST IS PASSED BACK TO SOFTWARE
      - 10MS TIMEOUT APPLIES TO CONTROL UNIT AND SHARED DASD/HEAD OF STRING BUSY ONLY

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

- 3081 I/O QUEUING (CONT'D)
  - I/O INTERRUPT
    - DATA STORED IN LOCATION 185 WHEN CSW IS STORED
      - DELAY CODE
        - BUSY CONDITION ENCOUNTERED
      - CHANNEL Q LENGTH
        - NUMBER OF REQUESTS QUEUED FOR THIS CHANNEL

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

CHANNEL UTILIZATION

- HIGH CHANNEL UTILIZATION AFFECTS QUEUE TIME AND SERVICE TIME
  - HAS GREATEST IMPACT ON SERVICE TIME
- RULE OF THUMB:
  - DASD CHANNELS SHOULD NOT EXCEED 35% UTILIZATION
- REFINEMENTS TO THE GENERAL RULE
  - IMS - 30%
  - TSO - 35%
  - BATCH - 40%

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

CONFIGURATION GUIDELINES

- ISOLATE DEVICES THAT SERVICE A GIVEN SUBSYSTEM TO THEIR OWN CHANNEL(S) AND CONTROL UNITS
  - ALLOWS INSTALLATION TO MONITOR AND TUNE I/O TO SPECIFIC SUBSYSTEM REQUIREMENTS WITHOUT EFFECTING OTHER SUBSYSTEMS

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

CHANNEL UTILIZATION

- SEQUENTIAL ACCESS
  - READ DIRECT VS READ PREVIOUS
    - OPTCD=Z
    - FBS DATA SETS
    - SAM-E (SU9)

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

CHANNEL CONFIGURATIONS

- OPTIMIZE PERFORMANCE, I.E., MINIMIZE OVERRUNS
- 303X CHANNEL CONFIGURATION GUIDELINES TECHNICAL BULLETIN (GG22-9020)

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

DEVICE UTILIZATION

- MEASURE OF DEVICE 'GOODNESS' OR 'BADNESS' IS RESPONSE TIME (QUEUE TIME PLUS SERVICE TIME)
- HIGH DEVICE UTILIZATION DOES NOT IMPLY HIGH SERVICE TIME HOWEVER, HIGH DEVICE BUSY CAN CAUSE EXCESSIVE QUEUE TIME

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

CONTROL UNIT CONTENTION

- CONDITION CODE SETTINGS

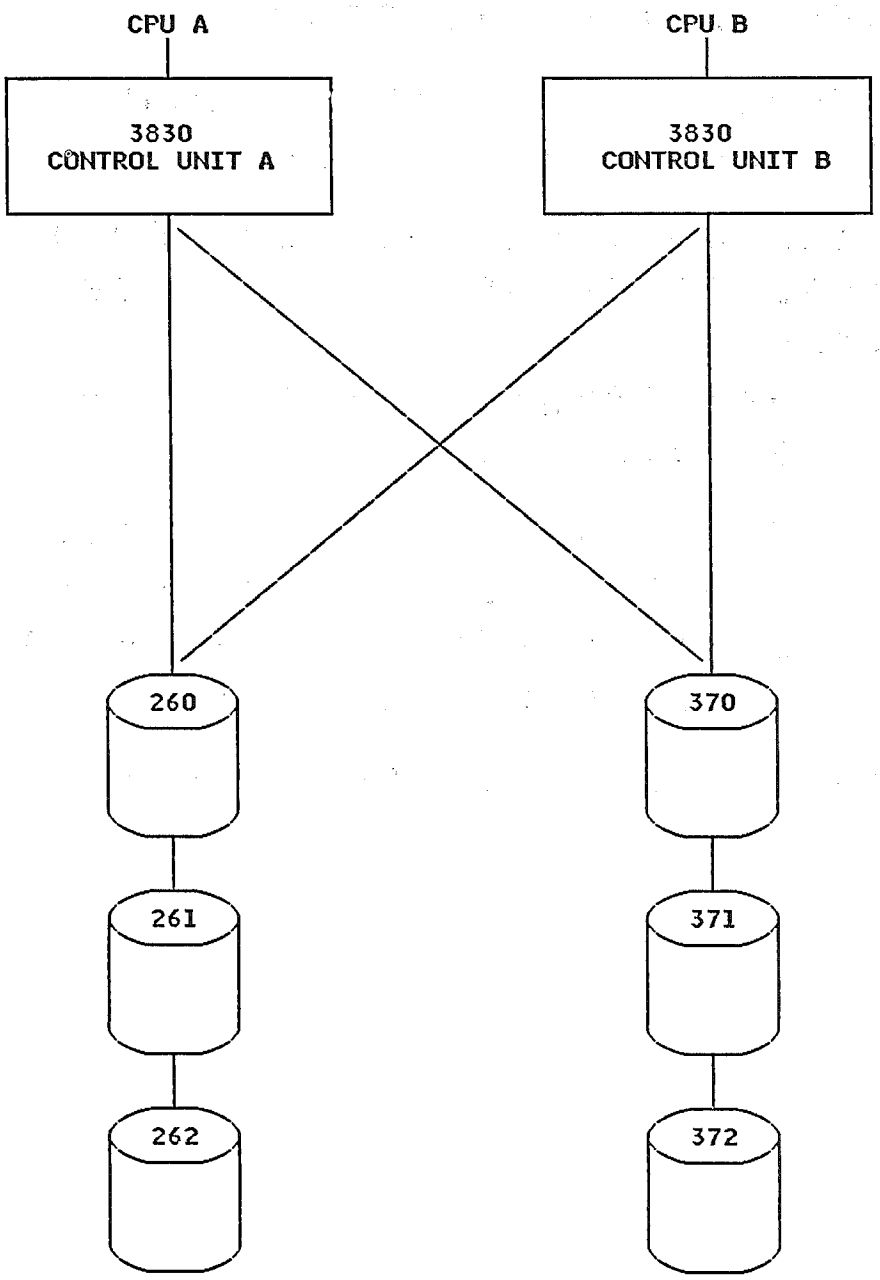
SIO CONDITION CODE	CSW	UCBBSY	UCBCUB	RMF
0	0	1	0	% DEV BUSY
1	BUSY & SM	0	1	% CU DELAY
1	BUSY & NO SM	1	1	% RESV DELAY

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

CONTROL UNIT CONTENTION

- QUEUE TIME DELAYS REPORTED BY RMF AS % CU DELAY
- HIGH CONTROL UNIT CONTENTION AFFECTS QUEUE TIME AND SERVICE TIME
  - HAS GREATEST IMPACT ON SERVICE TIME
- CONTROL UNIT BUSY GENERALLY CAUSED BY TOO MANY OR HIGH ACTIVITY DEVICES ATTACHED TO A SINGLE CONTROL UNIT
- RMF GENERALLY PROVIDES REQUIRED DATA

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

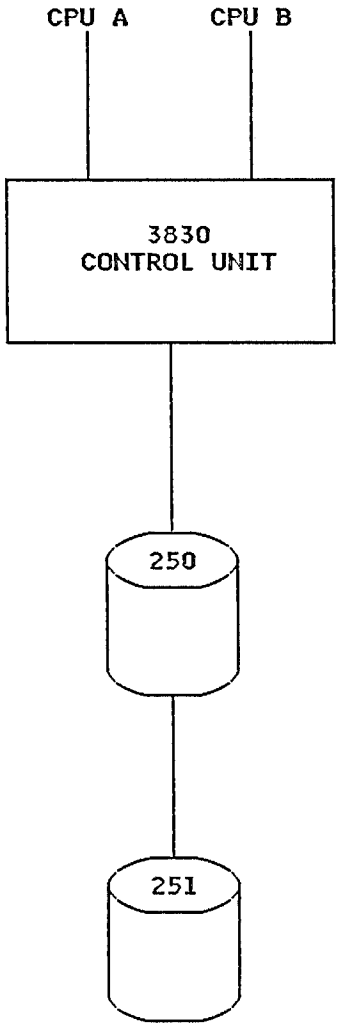


AN MVS TUNING PERSPECTIVE
I/O CONTENTION

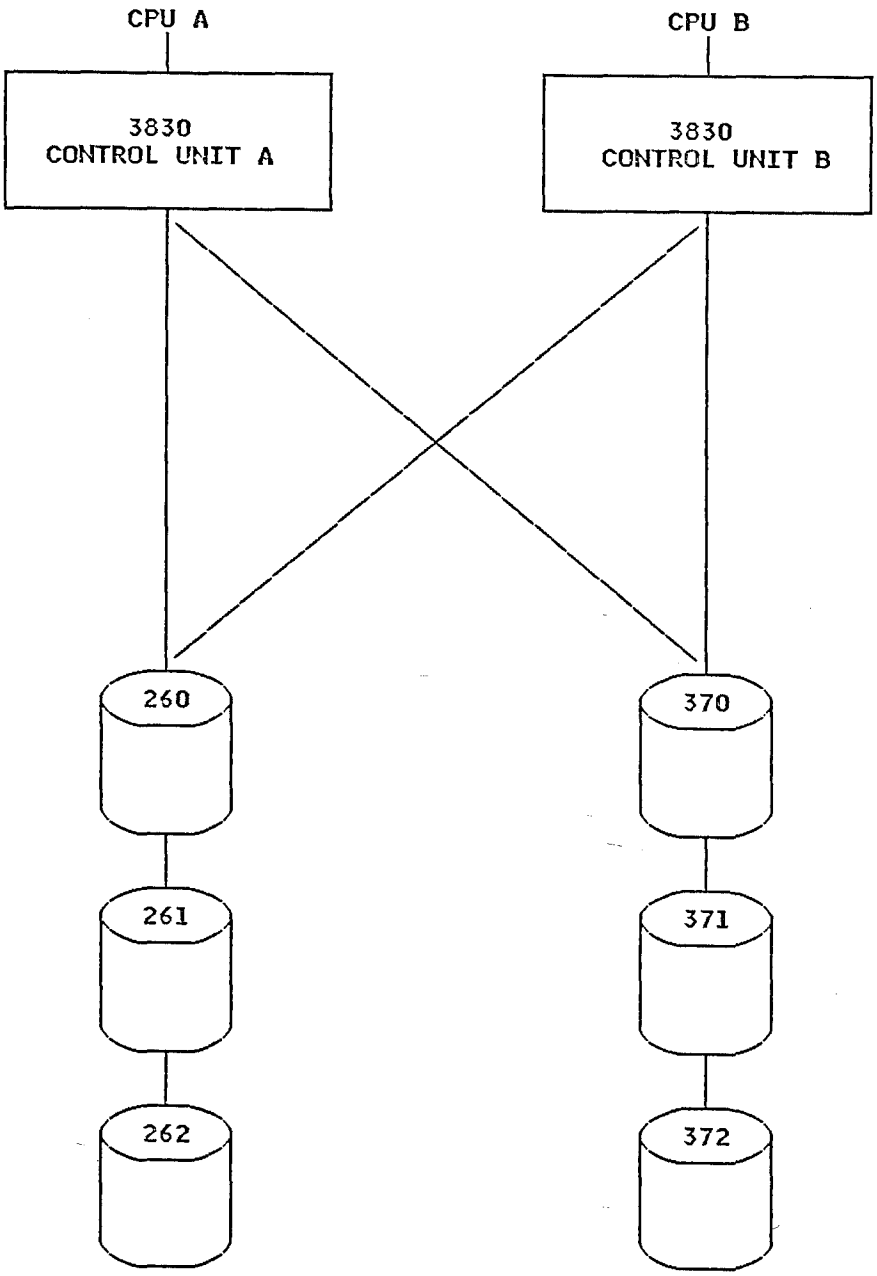
SHARED DASD/HEAD OF STRING CONTENTION

- REPORTED BY RMF AS % RESV DELAY
- CAN BE SIGNIFICANT IMPACT ON QUEUE TIME
- USUALLY INDICATES SHARED DASD CONTENTION

AN MVS TUNING PERSPECTIVE
I/O CONTENTION



AN MVS TUNING PERSPECTIVE
I/O CONTENTION



AN MVS TUNING PERSPECTIVE
I/O CONTENTION

SHARED DASD CONSIDERATIONS

- SHARE ONLY THOSE DEVICES THAT REQUIRE CONCURRENT ACCESS FROM MULTIPLE PROCESSORS
  - SPOOL, CATALOGS, PROGRAM LIBRARIES, ETC.
- NEVER SHARE WORK PACKS OR PAGING DEVICES
- ISOLATE SHARED DEVICES TO THEIR OWN CONTROL UNITS
  - ISOLATE SHARED DEVICES SO THAT THEY DO NOT AFFECT OTHER DEVICES CONTAINING NON-SHARED DATA
- ADVANTAGES
  - MONITOR AND CONTROL SHARED DASD
  - CORRECTIVE ACTION WITHOUT EFFECTING OTHER SUBSYSTEMS
- ADDITIONAL DASD REFERENCE
  - DASD CONFIGURATION AND SHARING CONSIDERATIONS (GG22-9052)

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

ARM CONTENTION

- AFFECTS QUEUE TIME AND SERVICE TIME
- CAUSED BY TOO MANY ACTIVE DATA SETS ON A GIVEN VOLUME AND/OR BY EXCESSIVE SEEK DISTANCES
  - EXAMINE RMF'S OPEN DATA SET COUNT FIELD
  - AVERAGE SEEK DISTANCE SHOULD NOT EXCEED 40 TO 50 CYLINDERS

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

PAGING CONFIGURATION TUNING (NON MVS/SE SYSTEM)

- EXPERIENCE CLEARLY INDICATES THE MOST RELIABLE WAY TO CONTROL MPL ADJUSTMENT IS VIA PAGE FAULT RATE
  - CPU, ASMQ, AND UIC SRM CONSTANTS SHOULD BE DEACTIVATED
  - PAGE FAULT RATE SRM CONSTANTS SHOULD BE ACTIVATED
- TUNING A PAGING CONFIGURATION AMOUNTS TO DEFINING THE NUMBER AND PLACEMENT OF PAGE DATA SETS TO EFFICIENTLY HANDLE THE PAGE RATE DICTATED BY THE PAGE FAULT RATE CONSTANTS

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

PAGING CONFIGURATION TUNING (NON MVS/SE SYSTEM)

- TUNING STEPS

  1. SET SRM PAGE FAULT CONSTANTS AS FOLLOWS:

PROCESSOR	PTR HIGH	PTR LOW
3031/158	25	20
3032/168	35	30
3033/168MP	55	50

     - INITIAL VALUES THAT MAY NEED ADJUSTMENT
     - GENERALLY MEASURED AGAINST INTERACTIVE OR ON-LINE SUBSYSTEM RESPONSE
     - SUBSYSTEMS INCURRING MORE THAN 5 TO 8 PAGE FAULTS PER SECOND GENERALLY HAVE RESPONSE PROBLEMS
       - 2 TO 3 IMS CONTROL REG
       - 3 TO 4 IMS MSG REGION

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

PAGING CONFIGURATION TUNING
(NON MVS/SE SYSTEM)

- TUNING STEPS

  2. MONITOR UTILIZATION

     - AVERAGE PAGE DATA SET SERVICE TIME 50 TO 70 MILLISECONDS

       - KEEP 3330/3350 PAGING DEVICES LESS THAN 35% BUSY
       - KEEP PAGING CHANNEL UTILIZATION BELOW 35%
       - MINIMIZE % CU DELAY
       - ZERO % RESV DELAY

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

PAGING CONFIGURATION TUNING
(NON MVS/SE SYSTEM)

- TUNING STEPS (CONT'D)
  - ADDITIONAL CONSIDERATIONS
    - PAGING VOLUMES SHOULD BE DEDICATED, I.E., NO OTHER ACTIVE DATA SETS ON SAME VOLUME
    - SWAP DATA SETS LESS SENSITIVE TO ARM MOVEMENT. 3330/3350 OK.
    - 2305 FOR PLPA AND/OR LOCAL PAGE DATA SETS

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

PAGING CONFIGURATION TUNING

- MVS/SE REL.1 SYSTEM
  - USE EXISTING PAGE FAULT RATE CONSTANTS (PTR)
  - CHANGE DEMAND PAGING RATE (DPR) AND/OR MILLISECONDS-PER-PAGE (MSPP)
- MVS/SE REL. 2 SYSTEM
  - STORAGE ISOLATION
    - WORKING SET
    - PAGE-IN RATE
  - SEE MVS/SE REL.2 PLANNING INFORMATION (GG22-9028) FOR RECOMMENDATIONS

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

PERFORMANCE ANALYSIS METHODOLOGY

- SCAN RMF DEVICE ACTIVITY REPORT
  - NOTE ALL DEVICES THAT APPEAR TO HAVE EXCESSIVE AVG Q LNGTH AND SERVICE TIMES
- CALCULATE RESPONSE TIME FOR EACH DEVICE NOTED ABOVE
  - QUEUE TIME PLUS SERVICE TIME
- ORDER DEVICES BY RESPONSE TIME
  - HIGHEST RESPONSE TIME FIRST
  - PRIORITY LIST
- ADDITIONAL MEASUREMENTS WHERE MORE DATA IS NEEDED
  - GTFPARS
- SERVICE LEVEL REPORTER PROGRAM PRODUCT (5740-DC3)
  - INPUT FROM SMF, RMF, IMS LOG
  - MULTIPLE SYSTEM INPUT
  - CALCULATE RESPONSE TIMES
  - USER DEFINED REPORTS

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

TUNING DATA SOURCES

- MEASUREMENT TOOLS
  - RMF
    - CHANNEL AND DEVICE UTIL.
    - CHANNEL/DEVICE SERVICE TIME
    - AVG Q LNGTHM
    - CONTROL UNIT DELAYS
    - DEVICE BUSY/RESERVE DELAYS
  - GTFPARS (PROG # 5798-CQQ)
    - GTF ANALYSIS TOOL
    - DETAILED I/O ANALYSIS
      - SERVICE TIMES
      - SEEK ANALYSIS
      - CONDITION CODE ANALYSIS

AN MVS TUNING PERSPECTIVE
I/O CONTENTION

OVER INITIATION

- PERFORMANCE IMPACT COMES FROM:
  1. CPU CYCLES REQUIRED TO MANAGE ADDITIONAL ADDRESS SPACES
  2. GREATER DEMANDS ON SPOOL AND WORK PACKS
  3. GREATER DEMAND FOR REAL STORAGE THAT RESULTS IN:
     - ADDITIONAL CPU CYCLES FOR PAGING AND SWAPPING
     - INCREASED PAGING CHANNEL UTILIZATION
     - HIGHER PAGING DEVICE UTILIZATION
- CAN'T DO MUCH ABOUT ITEM 1, BUT YOU CAN DO SOMETHING ABOUT ITEM 2 AND 3
  - TUNE PAGING CONFIGURATION

AN MVS TUNING PERSPECTIVE
SATURATED CPU

- RUNNING AT OR NEAR 100% UTILIZATION AND PERFORMANCE OBJECTIVES AREN'T BEING MET
- OPTIONS
  - FASTER PROCESSOR
  - TUNING EFFORT
  - CHANGE OBJECTIVES
  - ALL OF THE ABOVE

AN MVS TUNING PERSPECTIVE
SATURATED CPU

- POTENTIAL CAUSES
  - POORLY BLOCKED DATA SETS
  - EXCESSIVE USE OF SUPERVISOR SERVICES

AN MVS TUNING PERSPECTIVE
SATURATED CPU

- I/O BLOCKSIZES
  - GREATEST POTENTIAL FOR REDUCING CPU CYCLES
  - FIRST PLACE TO START
  - EXAMPLE OF FILE UPDATE PROGRAM WITH 4 QSAM FILES

BLOCKSIZE	200	1000	2000	4000	6000
EXCP'S	105,000	21,000	10,500	5250	3500
SIO'S	51,319	10,468	5277	2729	1800
ELAPSED SECONDS	600	150	103	71	56
CPU SECONDS	73.6	21.6	14.3	10.6	9.3
CHANNEL SECONDS	95.2	46.9	38.4	36.5	33.3

  - RESULTS OF INCREASING ALL BLOCKSIZES FROM 200 TO 6K
    - 87% REDUCTION IN CPU TIME
    - 90% REDUCTION IN ELAPSED TIME
    - 65% REDUCTION IN CHANNEL SECONDS

AN MVS TUNING PERSPECTIVE
SATURATED CPU

- I/O BLOCKSIZES
  - BLOCKSIZE RECOMMENDATIONS

    FOR SEQUENTIAL DATA (SAM)
    - NO BLOCKSIZE LESS THAN 4K
      - TAPE OR DISK
    - DISK BLOCKSIZES
      - 6K FOR COMPATABILITY WITH 3330/3350 DEVICES
      - FULL TRACK WHERE APPROPRIATE
    - TAPE BLOCKSIZES
      - 8 TO 12K REASONABLE
      - 32K WITH 6250 TAPES

AN MVS TUNING PERSPECTIVE
SATURATED CPU

- I/O BLOCKSIZES (CONT'D)
  - BLOCKSIZES VS PAGING

	2K BLKSIZE	12K BLKSIZE	PERCENT CHANGE
ELAPSED SECONDS	650	591	- 9.1
CPU SECONDS	608	537	-11.7
PAGES/SECOND	24.57	35.20	+43.3

  - IF PAGING CONFIGURATION IS TUNED, CPU CYCLE SAVINGS FROM INCREASED BLOCKSIZES MORE THAN COMPENSATES FOR COST OF HIGHER PAGING

AN MVS TUNING PERSPECTIVE
SATURATED CPU

- I/O BLOCKSIZES (CONT'D)
  - WHERE TO START
    - VIO FOR DATA SETS WITH 2K OR LESS BLOCKSIZES
    - PROCLIBS
  - NEXT
    - SMF
    - REDUCTION PROGRAM
      - DATA SET NAME
      - VOLUME SERIAL
      - DEVICE TYPE
      - BLOCKSIZE
      - NUMBER OF EXCP'S

AN MVS TUNING PERSPECTIVE
SATURATED CPU

- I/O BLOCKSIZES (CONT'D)
  - RECENT SURVEY OF 265,613 USER DATA SETS (47 MVS INSTALLATIONS) SHOWED:
    - 65 TO 70% OF USER DATA SETS ARE SEQUENTIAL (SAM)
    - SEQUENTIAL BLOCKSIZE DISTRIBUTION
      - 85% ARE 4K OR LESS
      - 70% ARE 2K OR LESS
      - 40% ARE 500 OR LESS

AN MVS TUNING PERSPECTIVE
SATURATED CPU

- SUPERVISOR SERVICES
  - GENERALLY LESS POTENTIAL THAN BLOCKSIZES FOR SAVING CPU CYCLES BUT WORTH INVESTIGATING
  - OBJECTIVE IS TO IDENTIFY APPLICATIONS USING EXCESSIVE NUMBER OF SERVICES
  - GTFPARS

AN MVS TUNING PERSPECTIVE
SATURATED CPU

- SUPERVISOR SERVICES (CONT'D)
  - GTFPARS ANALYSIS
    - GTF DATA COLLECTION
      - MOST CRITICAL STEP
      - SAMPLE PERIODS
      - ONE REEL OF TAPE PER SAMPLE
    - WHAT TO LOOK FOR
      - SVC RATE NO GREATER THAN:

PROCESSOR	SVC'S PER SECOND	EXCP/EXCPVR PER SECOND
3031/158	500	100
3032/168	1200	300
3033 *	2000	500

* ESTIMATES

AN MVS TUNING PERSPECTIVE
STORAGE CONTENTION

- SOLUTION TO STORAGE CONSTRAINED SYSTEM IS MORE REAL STORAGE
  - ASSUMES TCAM, VTAM, VSAM, IMS, ETC. BUFFERS ARE TUNED
- COMPLEX SYSTEM CENTER TECHNICAL BULLETIN (GG22-9053) EXAMPLE OF 168 MOVING FROM 5 TO 6 MEGS. (AVERAGE OF 2 SYSTEMS)
  - 43% DROP IN PAGING RATE
    - 43.3 TO 24.8 PAGES/SEC
  - 25% INCREASED BATCH THRUPUT
    - 53 TO 66.5 JOBS/HR
  - 33% DROP IN AVG TSO RESPONSE
    - 4.4 TO 2.9 SECONDS
  - 20% DROP IN AVG IMS RESPONSE
    - 5 TO 4 SECONDS
- CAN TUNE SYSTEM FOR OPTIMUM PERFORMANCE IN EXISTING STORAGE

AN MVS TUNING PERSPECTIVE
SUMMARY

PERFORMANCE MANAGEMENT

- PERFORMANCE CRISIS IS GENERALLY AN INDICATION OF A BREAKDOWN OR LACK OF PERFORMANCE MANAGEMENT
- SOME ELEMENTS OF A PERFORMANCE MANAGEMENT SYSTEM
  - DOCUMENTED PERFORMANCE OBJECTIVES
  - PRIORITIZED WORKLOADS
  - PERFORMANCE MONITORING

AN MVS TUNING PERSPECTIVE
SUMMARY
I/O CONTENTION

- RESPONSE TIME IS KEY
  - QUEUE TIME
  - SERVICE TIME
- ISOLATE I/O SUBSYSTEMS
- RULES OF THUMB
  - MINIMIZE QUEUE TIME
  - DASD CHANNEL UTILIZATION SHOULD NOT EXCEED 35%
    - IMS - 30%
    - TSO - 35%
    - BATCH - 40%
  - 3330/3350 DEVICE SERVICE TIME GENERALLY 25 TO 40 MILLISECONDS
  - AVERAGE SEEK DISTANCE SHOULD NOT EXCEED 40/50 CYLINDERS

AN MVS TUNING PERSPECTIVE
SUMMARY

TUNING PAGING CONFIGURATIONS

- SET APPROPRIATE SRM PAGE FAULT RATE CONSTANTS. MEASURE AND MODIFY, IF NECESSARY.
- MONITOR UTILIZATION
  - PAGE DATA SET SERVICE TIME 50 TO 70 MILLISECONDS
  - CHANNELS LESS THAN 35%
  - PAGING DEVICES LESS THAN 35%
  - CHECK AVG Q LGN, % CU DELAY, % RESV DELAY
- OTHER CONSIDERATIONS
  - DEDICATE PAGING VOLUMES
  - 2305 FOR PLPA AND/OR LOCAL PAGE DATA SETS
  - SWAP DATA SETS ON 3330/3350

AN MVS TUNING PERSPECTIVE
SUMMARY

SATURATED CPU

- I/O BLOCKSIZES GREATEST POTENTIAL FOR CYCLE SAVINGS
- SUPERVISOR SERVICES NEXT BEST

STORAGE CONTENTION

- SOLUTION IS MORE STORAGE
- CAN TUNE TO EXISTING STORAGE

READER'S COMMENT FORM

Title: An MVS Tuning Perspective
Washington Systems Center
Technical Bulletin GG22-9023-04

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Please state your occupation: _____

Comments:

Please mail to: R. M. Schardt
IBM Washington Systems Center
Field Support
18100 Frederick Pike
Gaithersburg, Md. 20760

GG22-9023-04

An MVS Tuning Perspective Printed in U.S.A. GG22-9023-04

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601