

PILOT TECHNIQUES

**Axios Products, Inc
353 Veterans Highway, Suite 204
Commack, NY 11725
Sales/Administration: (800) 877-0990 info@axiosproducts.com
Technical Support: (631) 864-3666 support@axiosproducts.com
Telecopier (Fax): (631) 864-3693 www.axios.com**

Preface

This publication contains information necessary for the operation of PILOT, a family of proprietary program products used for performance management and capacity planning. It provides data processing managers, system programmers, and capacity planners with information required to use this product.

Information in this publication is subject to significant change.

THIS MANUAL IS PROVIDED FOR THE SOLE AND EXCLUSIVE USE OF THE CUSTOMER. THE MATERIAL CONTAINED IN THIS MANUAL IS CONFIDENTIAL AND SHOULD BE SO TREATED. COPIES MAY BE PURCHASED FROM AXIOS PRODUCTS, INC. ANY UNAUTHORIZED REPRODUCTION OF THIS MANUAL IS PROHIBITED.

Seventh Edition (November, 2007)

This edition applies to Version 1.7 of the PILOT program products and to all subsequent versions and modifications until otherwise indicated in new editions or newsletters.

© Copyright 1987-2007 Axios Products, Inc. All rights reserved.

Contents

Performance Management.	<u>1</u>
Introduction.	<u>1</u>
Response Time And Tuning.	<u>1</u>
Batch Turnaround.	<u>1</u>
Service Level Objectives.	<u>2</u>
System Monitoring.	<u>2</u>
Identifying Bottlenecks.	<u>2</u>
Solutions To Problems.	<u>2</u>
Reporting.	<u>3</u>
 Basic Tuning Tips.	 <u>5</u>
 Capacity Management.	 <u>9</u>
Introduction.	<u>9</u>
 Sample Capacity Plan Using PILOT.	 <u>15</u>
Introduction.	<u>15</u>
Stage 1 - Resource Reporting and Tracking.	<u>15</u>
Stage 2 - Service Levels.	<u>21</u>
Stage 3 - Workload Characterization.	<u>21</u>
Stage 4 - Forecasting.	<u>22</u>
 Index.	 <u>25</u>

Performance Management

Introduction

Performance management is geared to a different audience than capacity planning. The capacity planner usually has to communicate with many people within the organization. This includes those with little knowledge of computers and those with the most technical computer literacy in the organization. Performance managers interact mostly with middle managers. Also, this function must be more responsive to day-to-day activities and short term capacity planning (3-6 months).

Performance Issues:

- Response time and tuning
- Batch turnaround
- Service level objectives
- System monitoring
- Identifying bottlenecks
- Solutions to problems
- Reporting

Response Time And Tuning

Everyone in data processing knows that they need “good response time”. However, this cannot always be obtained. Tuning MVS is important. As the system’s capacity is reached, the impact of an untuned system is evident. New users to MVS are usually elated by MVS just running and forget (or may not know) that the default tuning parameters that come with MVS need to be customized. The same is true for CICS,

IMS, DB2 and other major subsystems. All of these subsystems can have a major impact on the rest of the system. A poorly written application, or files not properly defined, can result in poor response time. Although this section is not intended to be a performance cookbook, a brief discussion on multi-programming levels (MPL) and basic tuning techniques will be discussed later in this section.

The key to “good response time” is “consistency”. Users will work accordingly. If the user is doing something trivial, then the response time should not exceed the amount of time for the user to lose his trend of thought. This is the very basis of the numerous reports written on sub-second response time. However, if the user is generating a report on-line, then this function should not override the person who needs immediate responses.

Batch Turnaround

Although today more systems are on-line, application development programmers, corporate personnel, and others need reports to be generated; the paperless society is still in the future. The timeliness of these reports is extremely important throughout most organizations. This usually becomes a requirement, as well, in service level agreements. The tricky part of this demand is to not divert enough resources to impact the on-line users. A well tuned system does not suffer when the “HOT BATCH” job runs. Setting up classes and priorities of jobs is the

most effective way to provide consistent turnaround. The MVS Installation Customization Specification table (ICS) is one of the parameters in MVS to accomplish this task.

Service Level Objectives

Some installations tune MVS based upon the demand for service by the end user and their importance in the organization. This statement can be taken two ways. In the first we ask “Who is important today?” In the second, the area supervisors and their management agree on acceptable service levels and priorities. Once the priorities of workloads are set, tuning and politics are easier to manage as resources become tight. This should be done when the seas are calm, not during a storm of hurricane proportions. When done early, the crisis that follows may be easier to manage and energy will be spent on the solution of problems and not their causes.

The performance management function of reporting to management on the service level objectives is important. The need to know if the service levels are being met should be included in a performance management report. The PILOT products provide various response-related reports to satisfy this requirement. Note that it is not always necessary to report daily on every transaction. Only the “bread and butter” transactions (critical, high volume or large resource consumption) need to be reported on a daily basis. Remember, the performance management function is to solve problems and reach short-term goals.

System Monitoring

All of the components of the system need to be monitored and tracked, making bottleneck analysis easier. A common mistake is that the performance analyst and the capacity planner share the same report in one format. Both need the same data, but they need it on different levels. The performance analyst needs to look at “a lot of detail data”, while the capacity planner needs to look at the “big picture.”

It’s like trying to sit in front of a real-time monitor all day to “wait” for a problem to happen. You may react to a minor spike and not to the real bottleneck. PILOT can be used to find the problem when you are not standing in front of your real-time monitor.

Identifying Bottlenecks

The PILOT products allow you to look at a lot of detail quickly. The query functions in the DASD, WORKLOAD, CHANNEL, and CICS Response time spreadsheets allow you to focus on the problems. Also, the data can be sorted in many different ways to find the problem (multiple views are helpful in bottleneck analysis). Additionally, the data is there to create ad hoc reports for quick presentations. For example: you are asked at a Change Management meeting, “Why do you need to move the IMS data base off the empty page pack?” A few graphs on queues and response times will show the effect, and afterwards, the same reports can be generated to show the improvement.

Solutions To Problems

This is performance management’s most critical function. The solutions may not

always be quick nor can they be implemented in a timely fashion. However, keep in mind that once a solution is found it should be documented for future use. Most performance solutions are usually buried in a status report that gets filed, making it difficult to find if a problem with similar symptoms arises. Partial PILOT files, with special reports, can be saved on diskettes and marked for future use. This is one approach that works well.

Reporting

Throughout this document the reporting functions of PILOT have been presented in different forms. For performance management, PILOT can be the first step to the paperless society. Give your boss the RMF summary spreadsheet (an overview of your entire system), including the most active channels. The workload and TSO spreadsheets also provide a host of information.

Basic Tuning Tips

This section will address some basic tuning issues. The author assumes that the terms used in this section are understood. Also, the suggestions are general guidelines to help those who do performance tuning. The degree of multi-programming level (MPL) can make an untuned system yield erratic performance. Basically, the MVS Installation Performance Specification (IPS) table is where service level objectives and MPL's are defined, using domains.

Overall suggestions:

Keep the IPS as simple as possible:

1. Define only control performance groups.
2. Define the smallest number of performance groups necessary. Do not skip numbers; MVS creates larger type 72 records to fill in missing values. For example:

Defining 1,2,...10 is good. Defining 1,2,3,...10,80...86,90 is poor if you skip numbers and define performance groups you are not using.

3. Avoid using time slicing unless you fully understand it.
4. Keep the APG range simple and spread your workload across APG ranges. For example:

APG	F0-FF	System task
	E0-EF	System subtasks

80	TSO trivial
85	TSO medium
74-7F	TSO long
70-7F	Hot batch
	etc.

5. Define and use the Installation Customization Specification (ICS) table to put jobs in the correct performance group, thereby defining workloads within the system.
6. If possible, disable the set command in MVS so that operators do not change the performance groups of workloads.
7. Dispatching priorities should be assigned in logical order: MVS system tasks, MVS subsystems (on-line oriented) and then batch.
8. In a MVS/SP environment keep the number of workload levels defined in the IPS to a minimum. Four is more than enough for smaller machines, for example:

WL=(1,50,98,100) (refers to CPU utilization)

Also, smaller machines (4300 class) tend to reduce service to workloads with high utilization when the default IPS, distributed in the IPO installation process, is used. Service level objectives appear to just make matters worse (the system tends to thrash). For system tasks and TSO periods, one defines

SRV=(1000,1000,1000,1000) for objective 1 instead of SRV=(1000,100,75, 0).

The WL parameter was removed with MVS/ESA 4.2. Service Objectives are defined differently.

To provide good performance in TSO:

- Define a large Min and Max MPL for TSO trivial tasks.
- The medium running TSO transactions is where your Min and Max will determine the commitment of resources based upon the service objectives for the defined domain. As you increase this MPL, any workload (usually batch) with lower dispatching priority will have less available resources.
- Do not define more than four TSO periods. Three should be enough, unless you have a lot of long running TSO transactions.
- Be sure that the paging subsystem is tuned. A key to paging is the length of time it takes to transfer the page into storage, not the rate itself. However, as the Page Rate increases, the averaging of the page transfer time over long periods may skew results. Last, if Page packs are cache controllers, make sure the cache is turned off for the volume it resides on. The Auxiliary Storage Manger in MVS always uses the "BYPASS CACHE" algorithm. The only exceptions would be in a SMS environment (dataset level caching), or you have other system data sets on the

volume. (It is not uncommon with the density of the DASD devices today).

- If your system has expanded storage, increase the criteria migration age for VIO. Also, if the system is experiencing a large amount of page movement from real to expanded storage, an increase in the criteria age for batch jobs is required. This is accomplished differently in pre-MVS 4.2 operating systems.

MVS 4.3 Tuning

- Introduced in March 1993, there are various features not available in prior releases of MVS.
- SMF synchronization allows you to synchronize RMF, SMF across product lines to the same interval. The biggest benefit is to match performance statistics in jobs and subsystems (CICS, DB2, and HSM) with RMF intervals. This is an increase of MVS overhead for this process. Depending on you capacity, this maybe reduce the amount of CPU cycles to run applications. PILOT/MVS workload templates tracks MVS overhead, this makes it possible to measure the impact!
- The parameter "FIXCIDX" on the Domain keyword "DMN=" uses less MVS overhead to determine the domain's priority then the "DSRV" and ASRV parameters .

There are various sources of information on the subjects discussed in this section. CMG, SHARE, and other tuning newsletters.

Capacity Management

Introduction

Capacity management is a function within an organization to control the technical resources for present and future data processing requirements within the organization's financial constraints.

This function is needed in every information system area in one form or another. That is, the degree of commitment to this function is the key to its success. The larger the organization the more formalized this function needs to be. There is more information and methodology for this process than ever before. A good source of knowledge is the Computer Measurement Group, called CMG. Another group, not as management orientated is the SHARE group, which has multiple divisions addressing this topic (Management, CPE, and Modeling).

Regardless of the tools used, there are basic approaches that can lead to successful capacity planning. Usually, as a side benefit, the analysis discovers performance bottlenecks and you can address them separately with more detailed data. This is why the installations that are busy putting out fires and not planning can never catch up. They are in react mode and are not looking ahead. They do not know how their business is expanding. Data processing today can be the edge a company needs to compete; it is no longer a luxury.

The following methodology does not have to be fully implemented on the first attempt. Instead, stages can be implemented, and

goals, to refine the process, should be planned. In short, capacity management is a process of continual refinement.

The basic methodology for capacity management is:

- Define Service Levels
- Workload Characterization
- Resource Reporting and Tracking
- Forecasting
- Cost Justification

In order to implement the above methodology, the need must be recognized at a level high enough in the organization. In other words, an associate systems programmer may be able to produce a valid capacity plan for the organization. However, if the information is not communicated to the other areas of the organization, the work will be in vain. Ideally (and this is usually possible in large shops), the capacity planner:

- should be removed from the day-to-day problems of the data center.
- report directly, or not too far from, the financial decision maker of the data center (the one who approves the new equipment).

Define Service Levels

It is desirable to know what service levels are required by the end user. Often, service levels cannot be agreed upon because the end users do not understand the computer system.

Application design and implementation can be the root of unreachable goals. This is an issue that must be studied from a different perspective as a part of performance management. The objectives of obtaining these service levels are to:

- Provide consistent response times to the end user to meet their business schedules;
- Provide an accurate set of reports for different levels of management;
- Increase communication to refine service levels; and
- Negotiate and agree on service levels.

Response does not only apply to on-line systems; batch run turnaround is just as important. There are many installations that would like to run on-line for 24 hours a day, but need a window for additional processing. This trend is changing with the improved system software available today. This puts the burden and focus on the on-line system capacity to be up and running 24 hours a day. This requirement has contributed to the growth of capacity planning in more installations today. In general, there are three answers you would expect from the end user when presented with the question “Is your response time satisfactory?” General questions get general answers. Before meeting with end users it is recommended that you become familiar with the systems and business aspects of your installation. It is the job of the capacity planner to relate the “business elements” to computer resources, not the end users. PILOT can be used to identify the transactions that have the worst response times, which are major resource consumers. The user types in transactions or

chooses them from a menu. It’s helpful to have a head start on the problems the user may encounter. Reporting on key transactions and resource consumption can reduce the amount of time required to resolve problems reported by end users. This, in itself, should lead to good communications with the end user. Once that is established, then service level agreements can be made.

Workload Characterization

Enough cannot be said about the importance of workload characterization. This is where the “big picture” is seen. It is very easy for most people to understand a pie chart displaying the resources needed to run a particular system. Financial decision makers will feel that the technical staff is finally presenting data in a form that can be understood. An accepted approach successfully used today is:

- Define workloads
 1. In gross terms or computer software terms (for technical use only).
 2. In terms of business elements.
- Profile each workload.
 1. Volume usage.
 2. Resource usage.
 3. Chronological usage.

The PILOT Workload feature makes this step in capacity planning easy to accomplish. By defining workloads according to Performance Groups, workloads can be analyzed. An example of gross characterization is a sampling defining CICS, TSO, MVS, BATCH, and “other” as separate workloads. The next step is to define workloads as

business elements. For example, define financial systems, manufacturing, inventory, etc. This data is invaluable for validating models (forecasting). Tracking volumes of transactions and the resources they use are provided with PILOT and can be added into the workload spreadsheets easily for other workloads not yet supported. For example, if ADABAS is used, the information on the number of requests can be used for transaction volumes. Chronological usage helps to determine that you are not just analyzing a peak or valley of the business cycle. Both extremes must be recognized if they exist. This information is usually easy to obtain and should be considered in the analysis. If you are meeting the users' requirements during peak business periods, then forecasting can be done from this period.

Resource Reporting And Tracking

Reporting data for capacity planning is different than for performance management in that the high level of detail is not needed. Choosing typical days was once an acceptable method of gauging resource consumption. Today, PILOT can be used to summarize data. Focusing on peaks will ensure that you will have enough resources for the off-shift and under-utilized times. Monthly reports providing the following information are needed:

- Hardware: CPU, channel, memory, and DASD utilization
- Workload: gross and business resource usage.
- Volume reports for key applications.

A notebook or diskette of graphs and data accumulated over time is worth its weight in

gold. It becomes more important as you find yourself putting out fires and not planning. Getting a good, accurate reporting system in place is extremely important.

Forecasting

After workload characterization is completed, changes in the environment are inevitable. Changes can make a given workload seem completely different if the impact of those changes are not included in the capacity management methodology. What changes?:

- Growth in transaction volume (more work);
- Additional terminals added (increase the transaction rate);
- New features and functions in existing applications;
- New applications;
- New system software (a new release of CICS or IMS, DB2, etc.);
- New hardware (converting to 3390's).

Forecasting is the function for which some type of modeling must be done. The model is not a single solution, but part of a methodology that allows you to make confident predictions based upon the previous functions of the capacity methodology. While the techniques for modeling are different, the results achieved will be the same. PILOT provides a model which simulates the way an operating system dispatches work. The difficult part of modeling is in translating business requirements into hardware requirements. This should sound familiar; the converse was stated earlier. Methodologies for the two models included in PILOT should be reviewed. The simulator uses path lengths in order to insure that workload resource consumption is profiled in a generic form that can be modified to

accommodate changes to the environment. The following is a brief description of how to model changes using the simulator.

To model new applications or new functions and features of existing applications, the following procedure can be used:

1. Benchmark the programs (transactions) in a test environment;
2. Calculate the path length, I/O rate, and paging rate;
3. Calibrate the model with measured results;
4. Estimate the transaction volumes;
5. Add the path length to an existing workload if a new function is being added to an existing application. Add a new workload definition for a new application;
6. Run the model. The results will yield CPU utilization, impact on the I/O system, and provide response times for existing and new workloads. Parameters and different scenarios can be run until service levels and other criteria are met.

Cost Justification

It is difficult to estimate the savings an organization will receive from the capacity management function. Various sources claim justification is not needed today because the birth of the “Information Age” is upon us and the spiraling need for know-ledge ensures that the organization will need a larger computer next year. The preceding statement does hold some truth, but in a sea of despair, the capacity planner can control the amount of money necessary to provide the data processing requirements to run the organizations’ business.

For example, data centers that do not plan tend to replace equipment blindly without regard to current resource requirements and the life expectancy of the new computer system. The least that can be done to justify new hardware is to provide a forecast of expected equipment utilization that is based on the current rate of resource consumption. Then, compare the cost of the computer equipment to the cost and benefit of an application. The workload characterization ratios can provide a basis for comparing an organization’s application’s resource consumption to the function it provides at the level of running the business. Does a particular application function benefit justify the business expense?

These decisions are not made by capacity planners, but this demonstrates the need for the function with the type of information it can provide to help senior executives make budget decisions. With the type of information provided by PILOT, fiscal decisions can be made to decide when the budget money will be needed.

In summary,

- Provide forecasted utilizations of new equipment.
- Provide a resource vs. budget workload characterization report.
- Report on improvements to existing applications and growth trends.
- Recommend enough computer resources to meet the business requirements with sufficient future growth capacity included.
- Provide follow-up. Reports of actual vs. planned resource utilization.

In all of the reports mentioned above, always include the impact felt by the end user. The

end user will benefit the most from your planning. This will directly improve the company's efficiency. There are numerous reports on results obtained through productivity improvements based on response time management to make this case.

Lastly, the bottom line of capacity management, and this is where it is different from performance management, is: "capacity management avoids problems, it does not solve them."

Sample Capacity Plan Using PILOT

Introduction

This section describes a sample capacity plan. The first step in implementing a capacity plan is to be sure that the methodology is understood by management. The level of detail and reports suggested are dependent upon management's understanding of the concepts of computer resource consumption. A basic rule of thumb for the novice capacity planner is: the less management understands, the fewer reports and more graphs you should provide (the supporting reports and detailed reports should always be available).

The methodology in the section entitled "Capacity Management" will be used to implement the capacity plan. The plan is divided into four stages:

1. Resource reporting and tracking
2. Define service levels
3. Workload characterization
4. Forecasting

Stage 1 - Resource Reporting and Tracking

In this stage, you will collect the information on system resource consumption. The reports generated are not intended to identify bottlenecks (although they sometimes do, if the bottleneck is severe). Refer to the section titled "Performance Management" for further information. The four types of reports that will be generated through this system are:

1. Hardware: CPU, channel, memory, and DASD.
2. Workload: Gross usage for TSO, BATCH, and CICS.
3. Volume: CICS and TSO transaction volume
4. Response: CICS and TSO response times.

A simple tip for the novice: get the basic reports and then refine the process as stages are completed. Use a week's worth of data to produce weekly reports which can later feed the monthly reporting system.

Set up the SMF data using PILOT/SMF to ensure data integrity. Refer to the installation instructions for PILOT/SMF to complete your SMF data management system. The following assumptions are made for this example:

1. Hardware is a 4381-2 running SP1.3
2. Weekly SMF data is called: DSN=-BACKUP.SMFWKLY, which is a generation data group.

The following job, provided in the source library member PILOTRMF, can be used to extract RMF summary data on a weekly basis:

```

//PILOT JOB (ACCT#), 'PILOT INSTALL', CLASS=A, MSGCLASS=X
//*
//RESET      EXEC PGM=IDCAMS                RESET DATA SETS
//SYSPRINT   DD  SYSOUT=*
//SYSIN      DD  *
//          DELETE PILOT.*.PRN
//          DELETE PILOT.RMF4381.WKLY
/*
//READSMF    EXEC PGM=READSMF, REGION=2048K
//STEPLIB    DD  DSN=PILOT.V1M5.LOAD, DISP=SHR
//SYSPRINT   DD  SYSOUT=*
//SYSOUT     DD  SYSOUT=*
//EXCEPT   DD  SYSOUT=*
//SYSUDUMP   DD  SYSOUT=*
//SYSUT1     DD  DSN=KLM.SMFWKLY(0), DISP=SHR
//SYSUT2     DD  DSN=PILOT.RMF4381.WKLY, UNIT=SYSDA, VOL=SER=,
//          SPACE=(CYL,(20,10)), DISP=(,CATLG,DELETE)
//SYSIN      DD  *
****
****          GET RMF DATA      PILOT/SMF CONTROL CARDS
****
          ST=00000,E=99999,          SORT BY DATE/TIME TYPE  NO WEEKEND DATA
          SEL=(30,70-76), WEEKDAYS, NODUP
//SORTWK01   DD  SPACE=(CYL,(20,10)), RLSE), UNIT=SYSDA
//SORTWK02   DD  SPACE=(CYL,(10,4)), RLSE), UNIT=SYSDA
//SORTWK03   DD  SPACE=(CYL,(10,4)), RLSE), UNIT=SYSDA
//*
//*          THESE STEPS CREATE CAPACITY MANAGEMENT FILES TO DOWN LOAD
//*
//RMF1       EXEC PGM=READSMF, REGION=2048K, COND=(4,LT,READSMF),
//          PARM='SMFPC,HRLY,TSO,PI(01,11,02,12,03,13,04,14,05,15)'
//STEPLIB    DD  DSN=PILOT.V1M5.LOAD, DISP=SHR
//SYSPRINT   DD  SYSOUT=*
//SYSOUT     DD  SYSOUT=*
//EXCEPT   DD  SYSOUT=*
//SYSUDUMP   DD  SYSOUT=*
//SYSUT1     DD  DSN=PILOT.RMF4381.WKLY, DISP=SHR
//SYSUT2     DD  DUMMY
//SYSIN      DD  *
****
****          PRODUCE 1ST SHIFT RMF STATS  PILOT/SMF CONTROL CARDS
****
          ST=00000,E=99999,
          SEL=(70-76), WEEKDAYS, PEAKTIME=(0800,1700), TIME=M
//*
//*          THE FOLLOWING DD CARDS ARE FOR PILOT/MVS SMFPC PROGRAM
//*
//RMFLOG     DD  SYSOUT=*
//RMFOUT     DD  DSN=PILOT.RMF01WL1.PRN, DISP=(,CATLG,DELETE),
//          SPACE=(CYL,(1,1)), UNIT=SYSDA
//RMFTSO     DD  DSN=PILOT.TSO01WL1.PRN, DISP=(,CATLG,DELETE),
//          SPACE=(CYL,(1,1)), UNIT=SYSDA
//RMFWKLD    DD  DSN=PILOT.WKL01WL1.PRN, DISP=(,CATLG,DELETE),

```

```

//          SPACE=(CYL,(1,1)),UNIT=SYSDA
//RMFDASD  DD  DSN=PILOT.DAS01WL1.PRN,DISP=(,CATLG,DELETE),
//          SPACE=(CYL,(1,1)),UNIT=SYSDA
//RMFPAGE  DD  DSN=PILOT.PAG01WL1.PRN,DISP=(,CATLG,DELETE),
//          SPACE=(CYL,(1,1)),UNIT=SYSDA
//RMFCTL   DD  *
*
*          PILOT/MVS CONTROL CARDS
*
*          CPU  MODEL=4381-14
*
*  GROSS WORKLOAD DEFINITIONS DEFINED IN IEAIPS00 IN SYS1.PARMLIB
*
*          WKLD WL=(MVSPGN,0,5,8,9),OTHER=Y    JES,VTAM,TAPE LIB SYS
*          WKLD WL=(CICS,4,7)
*          WKLD WL=(TSO,2,12)
*          WKLD WL=(BATCH,1,3)
*          DEVICE SPAN=Y                ONLY ONE RECORD PER DEVICE
// *
// *          SECOND SHIFT STATISTICS
// *
//RMF2     EXEC PGM=READSMF,REGION=2048K,COND=(4,LT,READSMF),
//          PARM='SMFPC,HRLY,PI(01,11,02,12,03,13,04,14,05,15)'
//STEPLIB DD  DSN=PILOT.V1M5.LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//EXCEPT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  DSN=PILOT.RMF4381.WKLY,DISP=SHR
//SYSUT2   DD  DUMMY
//SYSIN    DD  *
****
****          PRODUCE 2ND SHIFT RMF STATS  PILOT/SMF CONTROL CARDS
****
          ST=00000,E=99999,
          SEL=(70-76),WEEKDAYS,PEAKTIME=(1800,2400),TIME=M
// *
// *  THE FOLLOWING DD CARDS ARE FOR PILOT/MVS SMFPC PROGRAM
// *
//RMFLOG   DD  SYSOUT=*
//RMFOUT   DD  DSN=PILOT.RMF01WL2.PRN,DISP=(,CATLG,DELETE),
//          SPACE=(CYL,(1,1)),UNIT=SYSDA
//RMFWKLD  DD  DSN=PILOT.WKL01WL2.PRN,DISP=(,CATLG,DELETE),
//          SPACE=(CYL,(1,1)),UNIT=SYSDA
//RMFDASD  DD  DSN=PILOT.DAS01WL2.PRN,DISP=(,CATLG,DELETE),
//          SPACE=(CYL,(1,1)),UNIT=SYSDA
//RMFCTL   DD  *
*
*          PILOT/MVS CONTROL CARDS
*
*          CPU  MODEL=4381-14
*
*  GROSS WORKLOAD DEFINITIONS DEFINED IN IEAIPS00 IN SYS1.PARMLIB

```

```

*
      WKLD WL=(MVSPGN,0,5,8,9),OTHER=Y      JES,VTAM,TAPE LIB SYS
      WKLD WL=(CICS,4,7)
      WKLD WL=(TSO,2,12)
      WKLD WL=(BATCH,1,3)
      DEVICE SPAN=Y                          ONLY ONE RECORD PER DEVICE

//*
//*      THIRD SHIFT STATISTICS
//*
//RMF3   EXEC PGM=READSMF,REGION=2048K,COND=(4,LT,READSMF),
//      PARM='SMFPC,HRLY,PI(01,11,02,12,03,13,04,14,05,15)'
//STEPLIB DD DSN=PILOT.V1M5.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//EXCEPT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT1  DD DSN=PILOT.RMF4381.WKLY,DISP=SHR
//SYSUT2  DD DUMMY
//SYSIN   DD *
****
****      PRODUCE 3RD SHIFT RMF STATS  PILOT/SMF CONTROL CARDS
****
      ST=00000,E=99999,
      SEL=(70-76),WEEKDAYS,PEAKTIME=(0000,0800),TIME=M

//*
//*      THE FOLLOWING DD CARDS ARE FOR PILOT/MVS SMFPC PROGRAM
//*
//RMFLOG  DD SYSOUT=*
//RMFOUT  DD DSN=PILOT.RMF01WL3.PRN,DISP=(,CATLG,DELETE),
//      SPACE=(CYL,(1,1)),UNIT=SYSDA
//RMFWKLD DD DSN=PILOT.WKL01WL3.PRN,DISP=(,CATLG,DELETE),
//      SPACE=(CYL,(1,1)),UNIT=SYSDA
//RMFDASD DD DSN=PILOT.DAS01WL3.PRN,DISP=(,CATLG,DELETE),
//      SPACE=(CYL,(1,1)),UNIT=SYSDA
//RMFCTL  DD *
*
*      PILOT/MVS CONTROL CARDS
*
*      CPU  MODEL=4381-14
*
*      GROSS WORKLOAD DEFINITIONS DEFINED IN IEAIPS00 IN SYS1.PARMLIB
*
      WKLD WL=(MVSPGN,0,5,8,9),OTHER=Y      JES,VTAM,TAPE LIB SYS
      WKLD WL=(CICS,4,7)
      WKLD WL=(TSO,2,12)
      WKLD WL=(BATCH,1,3)
      DEVICE SPAN=Y                          ONLY ONE RECORD PER DEVICE

/*
//*****END OF JOB *****

```

The sample job has three steps, one for each shift.

Notice that paging and TSO data is collected only for the first shift. Also, DASD is requested with the span option to provide only one record per device for the entire shift. We are more interested in volume requests than response times to the devices. The performance of the device is a separate issue addressed in this capacity plan. After the files are formatted, they can be downloaded to the PC, where reports and graphs can be generated.

The graphs available will be listed by template (spreadsheet) followed by a brief explanation of why the information is important. The LOTUS 1-2-3 distribution template names are used and each should be customized and saved to put your tracking system in place. There are equivalent templates in EXCEL. Refer to the installation instructions for more information.

Template Graph Usage

RMF01WL1	This template provides averages, by hour, across the time period into the spreadsheet. The date range of the data is displayed and provided in all graphs. This template provides a great deal of general information about the system.
CPU	Shows how the CPU resources are being used on an average for each hour.
ASID	The Address Space mix graph shows the ratio of

TSO	TSO users to batch jobs and started tasks. The TSO average response across all periods and the number of transactions shows transaction volume.
DPAGE	A graph of demand paging vs. swap rate will give an indication of memory usage.
CHAN	Up to 24 channels can be graphed. Choose the most active channels to track, but be aware that channel activity usually changes with each shift.

PAG01SP	This template provides detailed paging and memory usage information for the first shift. The other two shifts are not usually needed. If yours is a low paging environment, this graph can be omitted. Choose at least the total paging graph.
TOTAL	Shows how much and what type of paging your system is performing. VIO, swapping, common, demand, and total paging are graphed.

TSO01WL1	This template provides detailed information about TSO resource consumption. This template is drawn from the TSO daily spread-
----------	---

	sheet by changing the X axis from "time" to "date".	Reporting and graphs for CICS are also needed, but, during stage 1 of the capacity plan, only response time and volume information are needed for each production CICS region. Transaction detail data will be addressed in stage 2. Depending on the CICS monitor installed, the appropriate JCL should be used, as provided by PILOT/-CICS. Only the summary file and the summary template are needed in this stage. In this example using the IBM monitor, the detail file is a comment.
RSP	Response time for periods one and two by hour.	
NUMXT	A stacked bar graph showing which period transactions ended as well as the volume.	
WKL01WL1	The workload spreadsheet will summarize the gross usage of resources by each major system. It summarizes the total usage of each workload for the entire period of the data imported.	
CPU	CPU pie chart of utilization and a stacked bar graph showing CPU usage for each workload.	
TOTAL	Stacked bar graph of total service units for each workload.	
OVHD	Track your MVS overhead. You will notice an increase as capacity decreases and paging increases.	
DAS01SP	The DASD utilization of the packs across shifts is very useful.	
SIO	The total I/O's per string can be presented in a series of graphs.	
%BSY	The utilization of the packs can also be presented in a series of graphs.	

```

// P I L O T C I C   J O B
(ACCT#),'PILOT/CICS',MSGCLASS=X,
// CLASS=A
//* PILOT/CICS
//* SMFPC110 WILL FORMAT CMF
DETAIL RECORDS
//*
// S T A T S       E X E C
PGM=SMFPC110,REGION=1512K
// S M F I N           D D
DSN=KLM.BACKUP.CMFDAILY(0),DIS
P=SHR
//SMFLOG DD SYSOUT=*
// S M F 1 1 0 S           D D
DSN=PILOT.SUMCICS.PRN,      **
SUMMARY
/ /
SPACE=(TRK,(4,1),UNIT=SYSDA,DISP=(
,CATLG)
// * S M F 1 1 0 S R       D D
DSN=PILOT.RSPCICS.PRN,    ** DETAIL
/ / *
SPACE=(TRK,(4,1),UNIT=SYSDA,DISP=(
,CATLG)
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// S O R T W K 0 1       D D
UNIT=SYSDA,SPACE=(CYL,(10,1))
// S O R T W K 0 2       D D
UNIT=SYSDA,SPACE=(CYL,(10,1))

```



```
// S O R T W K 0 3      D D
UNIT=SYSDA,SPACE=(CYL,(10,1))
// S O R T W K 0 4      D D
UNIT=SYSDA,SPACE=(CYL,(10,1))
// S O R T W K 0 5      D D
UNIT=SYSDA,SPACE=(CYL,(10,1))
```

The data is down-loaded to the PC and the CICS summary spreadsheet is used to provide reports and graphs for response times and transaction volume.

Stage 2 - Service Levels

As mentioned in the “Capacity Management” section, service levels must be agreed upon by the end user and the MIS management. Availability of an on-line system, response time ranges for key transactions, and batch turnaround are a few of the items that can be used to measure service.

Availability data for an on-line system can be produced by either PILOT/SMF program RDSMF5 or PILOT/MVS program SMF-PC30. Both have similar output formats and differ only in the amount of information provided. PILOT/MVS provides more information and will format the data for down-loading to your PC for customizing your reports. Here is an example of using PILOT/MVS with PILOT/SMF to do the data reduction for the CICS regions by job name.

```
// R E A D S M F      E X E C
PGM=READSMF,REGION=720K
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//EXCEPT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// S Y S U T 1      D D
DSN=SMF.WEEKLY(0),DISP=SHR
```

```
// S Y S U T 2      D D
DSN=&&SMF30,UNIT=SYSDA,
DISP=(,PASS)
//SYSIN DD *
ST=00000,E=99999,SELECT=(30),
JOBNAME=(CICSP*,CICSMRO)
/*
// P C F M T      E X E C
PGM=SMFPC30,REGION=720K,
// COND=(4,LT,READSMF)
//SMFIN D      D
DSN=&&SMF30,DISP=OLD
//SMFLOG DD SYSOUT=*
// P R O D C I C S      D D
DSN=TSU01.T30CICS.PRN,
// DISP=(,CATLG),
// SPACE=(TRK,(4,4),RLSE)
//SMF30 DD DSN=TSU01.SMF30.PRN,
// DISP=(,CATLG),
// SPACE=(TRK,(4,4),RLSE)
//SMFCTL DD *
* COLLECT PRODCICS JOBSTATS
GROUP=(PRODCICS,CICSP1,CICSP2,
CICSP3,CICSMRO)
//
```

Stage 3 - Workload Characterization

In this stage, you will do a more detailed workload analysis. Reading the modeling methodologies and techniques section of this guide is a prerequisite for this stage in capacity planning. For this example, we will analyze a CICS system which is made up of multiple regions. This stage is a refinement of the first stage when we looked at our systems with gross measurements. The workload function of PILOT/MVS is used to profile the production workloads in fine detail. Identifying and reporting on key transactions used in CICS is also an important capacity issue. This must be done in

order to do forecasting (Stage 4) in terms of business elements. The transaction volumes become important once the relationship between the key transactions and business activity are understood. To start:

1. Run PILOT/MVS defining the desired workload's behavior during peak times. Ensure that you use at least a week's worth of data.
2. Run PILOT's Model Generator program against the same data. Here is an example:

```
//PILOT CICS JOB 'TEST
SIMBUILD','PILOT/CICS',
//
MSGLEVEL=(1,1),MSGCLASS=Q
//*
//* MODEL GENERATOR PRODUCES:
//*
//* 1. SIM FILE FOR THE SIMULATOR
//* 2. MODEL INPUT PARAMETERS FOR
MODLCICS SPREAD//*
SHEET WRITTEN TO SIMLOG
//*
//GENMODEL EXEC
PGM=SIMBUILD,REGION=1024K
//STEP LIB DD
DSN=PILOT.V1M5.LOAD,DISP=SHR
//SIMLOG DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SMFIN DD
DSN=BACKUP.SMFWKLY(0),DISP=SHR
//SIMOUT DD
DSN=PILOT.MODEL.SIM,DISP=SHR
>*.SIM
//SIMWKLD DD
DSN=PILOT.WKL01SIM.PRN,
//
DISP=(,CATLG),SPACE=(TRK,(4,4))
//* TMON FILE
//MONIN DD
DSN=TSU02.TLMK61,DISP=SHR TMON
```

```
//SIMCTL DD *
CPU MODEL=4381-2
PEAK TIME=(1400,1500),WKDAYS=Y
WKLD WL=(MVSPGN,0,9,10),PRTY=1
W K L D
WL=(CICSMRO,6),PRTY=2,APPLID=CIC
SMRO,
DISKS=(CICSPK,VSAM*)
W K L D
WL=(PRODCICS,7),PRTY=3,APPLID=CI
CS,
DISKS=(CICSPK,VSAM*)
//
```

The output of this job also contains the input parameters for the CICS Model spreadsheet. Refer to the PILOT/CICS User's Guide for more information on the modeling methodology.

3. Run and validate a model using the PILOT/CICS Simulator.
4. The reporting of key transaction's response times, volumes, and resource usage will profile the system. The PILOT/CICS's INCLUDE and EXCLUDE keywords are used to produce a detailed report by hour.

Stage 4 - Forecasting

Refer to the discussion on forecasting in the Capacity Management chapter of this section. That section describes the technique for modeling. This stage is an extension of the workload characterization process and addresses the need to retain "good response time." Remember, the capacity plan is a process which can be implemented and refined as the "what if" questions need to be answered.

Index

cache.	<u>6</u>
capacity planning.	<u>2, 1, 9-11, 21</u>
chan.	<u>19</u>
channel..	<u>2, 11, 15, 19</u>
cics.	<u>1, 2, 6, 10, 11, 15, 17, 18, 20-22</u>
cmf.	<u>20</u>
cpu.	<u>5, 6, 11, 12, 15, 17-20, 22</u>
daily.	<u>2, 19</u>
dasd..	<u>2, 6, 11, 15, 19, 20</u>
date..	<u>16, 19</u>
day.	<u>1, 2, 9, 10</u>
dd.	<u>16-18, 20-22</u>
default.	<u>1, 5</u>
end.	<u>2, 9, 10, 12, 13, 18, 21</u>
excel.	<u>19</u>
except.	<u>16-18, 21</u>
exclude..	<u>22</u>
form.	<u>9-11</u>
graph..	<u>19, 20</u>
group..	<u>5, 9, 15, 21</u>
hrly..	<u>16-18</u>
ics..	<u>2, 5</u>
include..	<u>12, 22</u>
interval..	<u>6</u>
jcl.	<u>20</u>
jobname.	<u>21</u>
keywords.	<u>22</u>
lotus.	<u>19</u>
methodology.	<u>9, 11, 15, 22</u>
model.	<u>11, 12, 17, 18, 22</u>
model generator.	<u>22</u>
page.	<u>2, 6</u>
paging.	<u>6, 12, 19, 20</u>
parameters.	<u>1, 2, 6, 12, 22</u>
parmlib.	<u>17, 18</u>
path length.	<u>12</u>
peaktime..	<u>16-18</u>
performance.	<u>2, 1-3, 5, 6, 9-11, 13, 19</u>
performance tuning.	<u>5</u>

pilot/cics.	20 , 22
pilot/mvs.	6 , 16-18 , 21 , 22
pilot/smf.	15-18 , 21
prn.	16-18 , 20-22
rdsmf5.	21
readsmf.	16-18 , 21
response.	1 , 2 , 10 , 12 , 13 , 15 , 19-22
response time.	1 , 2 , 10 , 13 , 20-22
rmf.	3 , 6 , 15-18
rmfctl.	17 , 18
rmfdasd.	17 , 18
rmflog.	16-18
rmfout.	16-18
rmfpage.	17
rmftso.	16
rmfwkld.	16-18
select.	21
service levels.	2 , 9 , 10 , 12 , 15 , 21
simbuild.	22
simulator.	11 , 12 , 22
smf.	6 , 15-18 , 21
smf30.	21
smfctl.	21
smfin.	20-22
smflog.	20 , 21
smfpc110.	20
smfpc30.	21
span.	17-19
spreadsheet.	3 , 19-22
stages.	9 , 15
start.	10 , 22
steplib.	16-18 , 22
string.	20
summary.	3 , 12 , 15 , 20 , 21
sysin.	16-18 , 21
sysprint.	16-18 , 21
system.	2 , 1-3 , 5 , 6 , 9-12 , 15 , 19-22
sysut1.	16-18 , 21
sysut2.	16-18 , 21
tasks.	5 , 6 , 19
template.	19 , 20
time.	1 , 2 , 5 , 6 , 10 , 11 , 13 , 16-22
transactions.	2 , 6 , 10-12 , 19-22
tso.	3 , 5 , 6 , 10 , 15-19

type. [5](#), [11](#), [12](#), [16](#), [19](#)
unit. [16-18](#), [20](#), [21](#)
utilization. [5](#), [11](#), [12](#), [20](#)
volume. [2](#), [6](#), [10](#), [11](#), [15](#), [19-21](#)
workload. [2](#), [3](#), [5](#), [6](#), [9-12](#), [15](#), [17](#), [18](#), [20-22](#)

