IBM

Washington
Systems
Center

Technical
Bulletin

Capacity Planning

An Introduction

by Dr. L. Bronner
Washington Systems Center

DAPS Code 0911
GG22–9001–00
January 1977

Washington Systems Center
Gaithersburg, Maryland
Technical Bulletin

An Introduction to Capacity Planning

by LeeRoy Bronner
Complex Systems Center
Gaithersburg, Maryland

This Technical Bulletin is being made available to IBM and customer personnel. It has not been subject to any formal review and may not be a total solution. The exact organization and implementation of the functions described will vary from installation to installation and must be individually evaluated for applicability.

A form is provided in the back for comments, criticisms, new data, and suggestions for future studies, etc.

DAPS Code 0911
GG22-9001-00
January 1977

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

List of Figures

1.0 Introduction

1.1 Capacity Planning Overview

Capacity Planning has become a critical topic for Data Processing (DP) executives. Their data processing systems have become very complex and the executives are encountering many problems in managing these systems.

The complexity is increased when there is a shift from a batch to an on-line system, from a single application environment to a multiple one (e.g., IMS, TSO, batch), and the migration to a different operating system or hardware configuration. The level of complexity increases when there is a multiple CPU environment with shared DASD.

To maintain their credibility, data processing executives must be able to provide reasonable answers and the associated analysis to the following types of questions.

- "HOW MUCH SHOULD I SPEND FOR DATA PROCESSING NEXT YEAR?"
- "WHAT HAPPENS TO MY ON-LINE SYSTEM WHEN I ADD 150 TERMINALS TO IT?"
- "CAN I INSTALL IMS/VS AND MAINTAIN CURRENT SYSTEM PERFORMANCE LEVELS?"
- "WHEN SHOULD I UPGRADE MY CPU?"
- "HOW MUCH MEMORY DO I NEED TO RUN IMS/VS, TSO, AND BATCH?"
- "AT WHAT LEVEL OF PERFORMANCE IS MY SYSTEM RUNNING TODAY?"
- "HOW MUCH CAPACITY DO I HAVE LEFT WHEN I AM RUNNING MY CPU AT 100% UTILIZATION?"

The data processing executive is searching for an approach to answering these types of questions.

What is Capacity Planning

Capacity Planning is the term used to identify a methodology for management and control of the complex DP environment. Capacity Planning addresses the problems involved in managing the computer resources, namely

- What parameters to collect to characterize the workload
- What parameters to collect to characterize the software and hardware components
- What tools are required to collect the data described in items above
- How the DP executive should manage his installation using this data.

It is basically a performance oriented approach to management. By this process, the loading, utilization and response of the various system resources are monitored and analyzed. Also, the flow of current and future work through the system is controlled to provide the best overall user satisfaction. User satisfaction is a very critical factor in the capacity planning process. Capacity planning is developed in general terms in this subsection with the supporting detail provided in later sections.

Capacity Planning Structure

To many people, capacity planning has meant only the collection of system performance data from which trends and projections are made. This is an important part of the capacity planning effort but without an organizing structure, this knowledge is only a mere collection of observations and conflicting incidents. Therefore, the initial step in the development of a capacity planning effort is the identification of this organizing structure (Figure 1). This structure forms a system of application areas. The systemization provides a basis for understanding the interrelationships of the various applications. Also, this structure will allow for a more effective interpretation and understanding of measured system parameters.

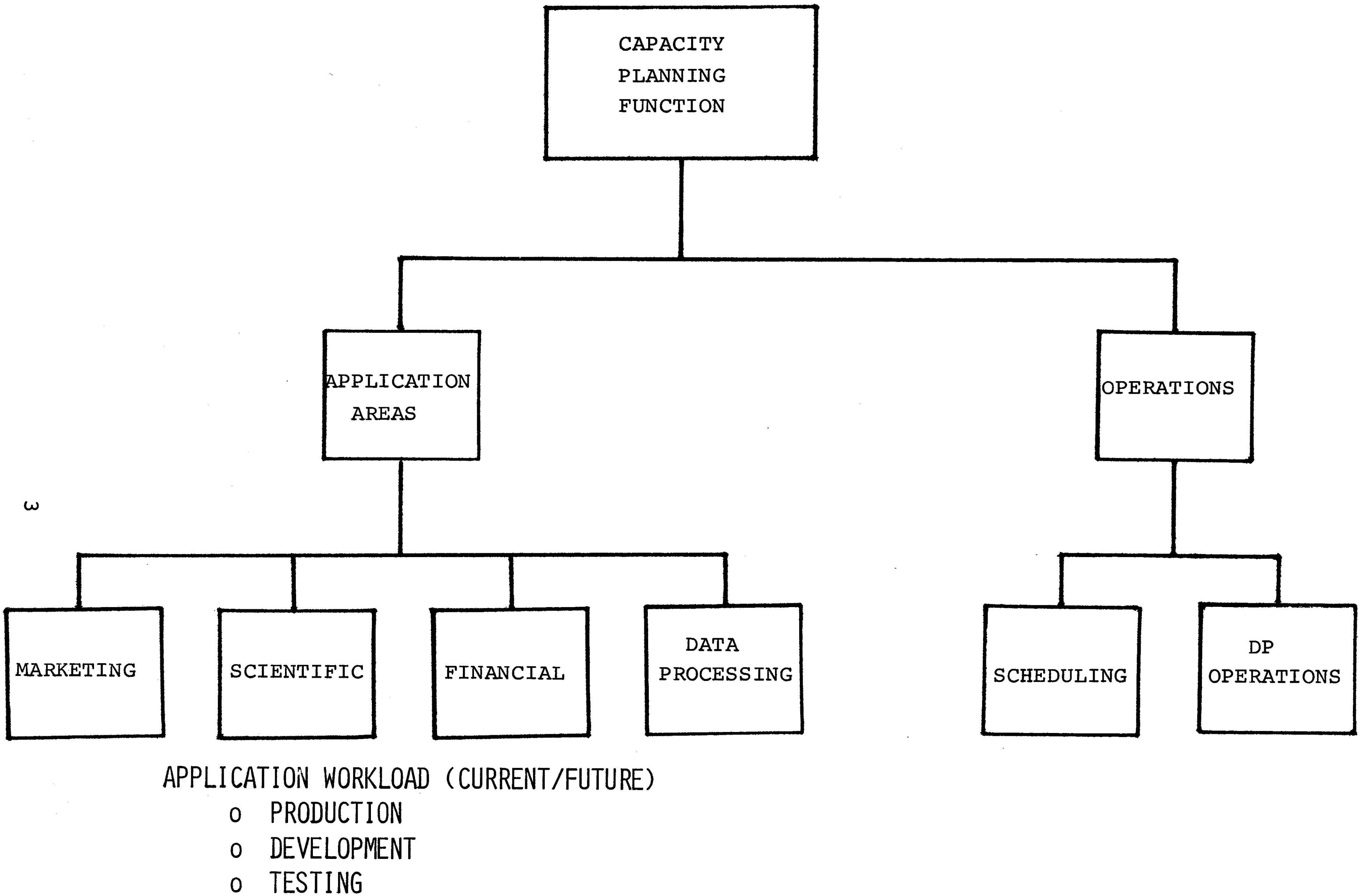


FIGURE 1. ORGANIZING STRUCTURE FOR CAPACITY PLANNING.

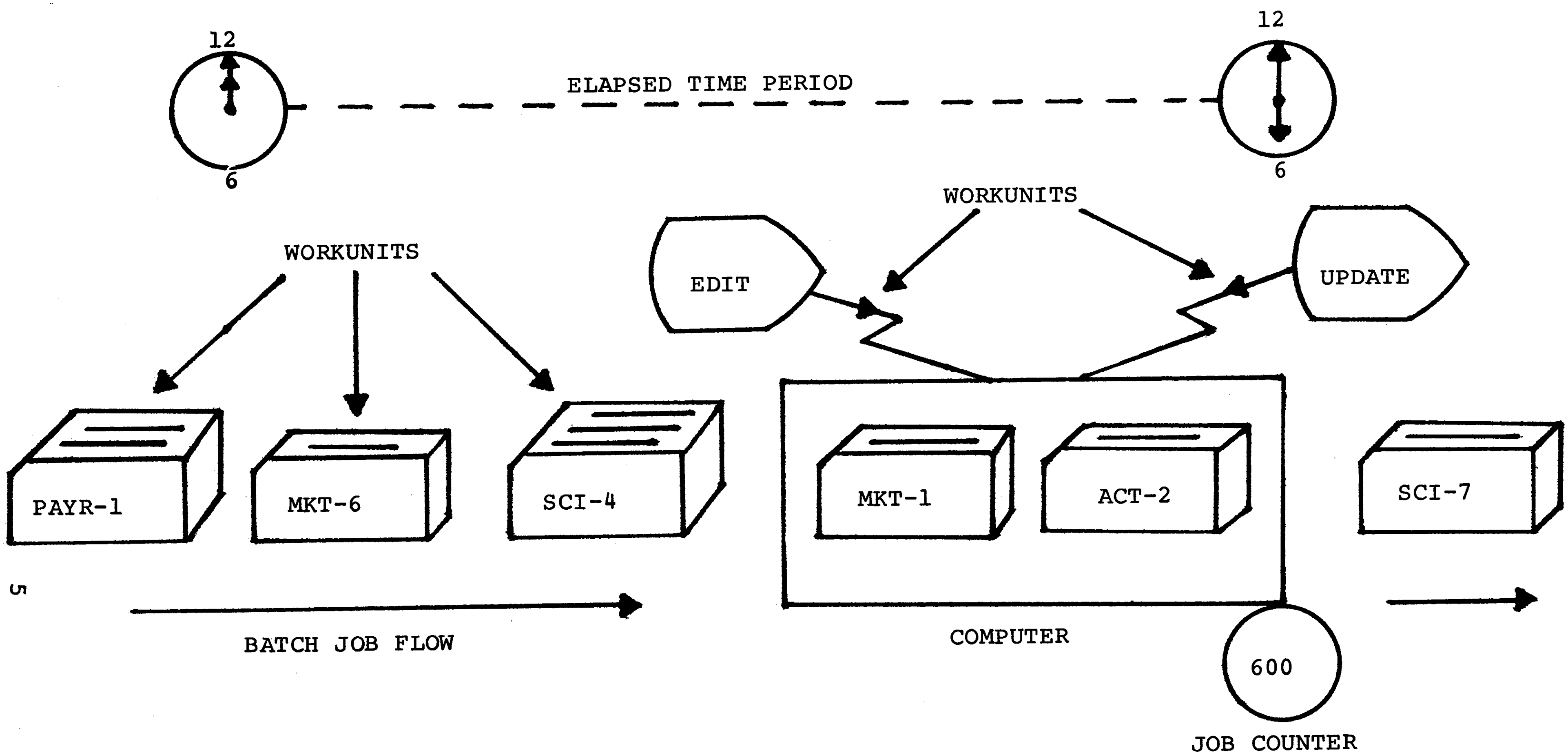
Definition of Workunits and Workload

One of the benefits derived from the capacity planning effort is an understanding of the current DP environment. Very crucial to this understanding is a segmentation of the data processing work to be accomplished into the units of work (workunits) and the load (workload) they place on the system (Figure 2). A workunit is defined as a externally generated fixed quantity of work to be accomplished by the computing system (e.g., 1 batch job, 1 inquiry, 1 command). The workload is the number of workunits processed by the system during a specific period of time (e.g., 100 batch jobs/hour, 10 transactions/second, 2 commands/second). In any given installation, it may be possible to define several types of workunits. Some workunits and workloads are listed below for various subsystems.

Subsystem	Workunit	Workload
Batch	Job	Jobs/Hour
CICS	Transaction	Transactions/second
IMS	Transaction	Transactions/second
TSO	Command	Commands/second
APL	Command	Commands/second

Segmentation By Application Area

As a means of developing a better understanding of system operation, the total workload should be segmented or accounted for by each major application area. Each workunit requires a certain amount of processor service (CPU hours). Therefore, identification of the workload by application areas will allow for segmentation of CPU service time. This approach to analysis develops a better understanding of DP operation, because, DP requirements (new projects, new loads, etc.) are normally estimated by application area. By measuring and understanding current workload, we are able to better understand future requirements. For example, if the current marketing application programs require 2 hours of daily CPU processing time and a new project is estimated as having the same complexity then the new program will require 2 hours of CPU. This becomes a good basis for estimation.



WORKUNIT - (1 BATCH JOB OR 1 TSO COMMAND)

$$\text{BATCH WORKLOAD} = \frac{\text{NUMBER OF BATCH JOBS PROCESSED (600)}}{\text{ELAPSED PERIOD OF TIME (6 HOURS)}}$$

$$= \text{100 BATCH JOBS PER HOUR}$$

FIGURE 2. DEFINITION OF WORKUNIT AND WORKLOAD.

Also, validation of these estimates are readily available once the applications are placed in production.

Capacity Planning Focal Point

Within the capacity planning function (Figure 1), a focal point is required. This point is the applications co-ordinator. Each application area is monitored from a system point of view. For example, in trying to match future software requirements to hardware characteristics, a system view of the applications is essential. Without an adequate systems view, the task of converting to a new hardware or software system becomes very difficult. The initial indications from the field are that many large installations have allowed various application areas to grow without maintaining system wide management. Normally, the various departments understand their application area in detail, but the system wide focus has been lost over the years. One of the reasons for this loss in system focus may be that the measurement tools necessary to provide this perspective have not been adequate. The measurement tool required to provide a system focus must allow resource utilizations to be broken down by application program and summarized by application area, and user service to be measured and summarized by application area.

Measurement Tools

Although current measurement tools are not as complete as most analysts would like, they will provide an initial basis for the development of a capacity planning program. SMF (System Measurement Facility) is currently the best tool available that allows resource utilization (CPU, channel, I/O device) to be segmented by job and summarized by application area. SMF should be understood by continuous system tracking and assessment. The primary issue is to obtain a gross feel for total resource consumption by application area. For example, does the financial application area appear to consistently consume about 15 SMF job-hours per week, 5% of a block multiplexor channel and 15% of two 3330 disk drives. Although SMF data is not complete, a capacity planning program initiated in this manner (i.e., organized structure, tracking, analysis) begins to offer insights into the daily operations of the installation not possible otherwise. The key to this initial phase is the continuous tracking and assessment.

SMF does not collect detail data on teleprocessing systems (i.e., IMS, CICS). Hence, additional tools must be used in conjunction with SMF data such as the CICS Performance Analyzer and the IMS/VS Monitor Report Print Program. Initially, it is very important to minimize the number of tools being used so that one will become proficient with the tools. Another tool that enhances this initial effort is the RMF (Resource Measurement Facility) available for MVS systems. RMF is the updated version of MF/1 (Measurement Facility 1).

Trends in Workload and Resource Utilization

The performance data collected should be graphed for analysis. Through these graphs, possible data trends can be identified; for example, a continual growth in CPU utilization by a given application area. At this point, one needs to begin to correlate various interacting factors such as an increase in transaction activity in the marketing area is manifest by an increased resource utilization (CPU, channels, etc.). Seeing some type of trend, gross projections may be made and validated during the next measurement period. This type of system interaction will increase your understanding of current operations and begin to establish a certain confidence in discussing future DP needs. Keep in mind, no complex modelling, queueing theory or any other analytical techniques have been discussed and we are already considering future DP requirements.

Projections for New Workloads

In the previous paragraph, it is suggested that gross trending and workload assessments will begin to provide some insight into future DP needs. Also, new projects are being implemented and placed in production. It is very critical to assess their impact on resource utilization. Then, along with this assessment, try to characterize the new job for comparison and estimation of other new projects to be implemented. This procedure is developing a base for future projections. In planning for the next years DP budget, where each department has outlined all new applications, program characteristics may be compared and projections made. Procedures are already in place where forecasts may be validated and projection policies altered if necessary. This type of cyclic procedure, namely

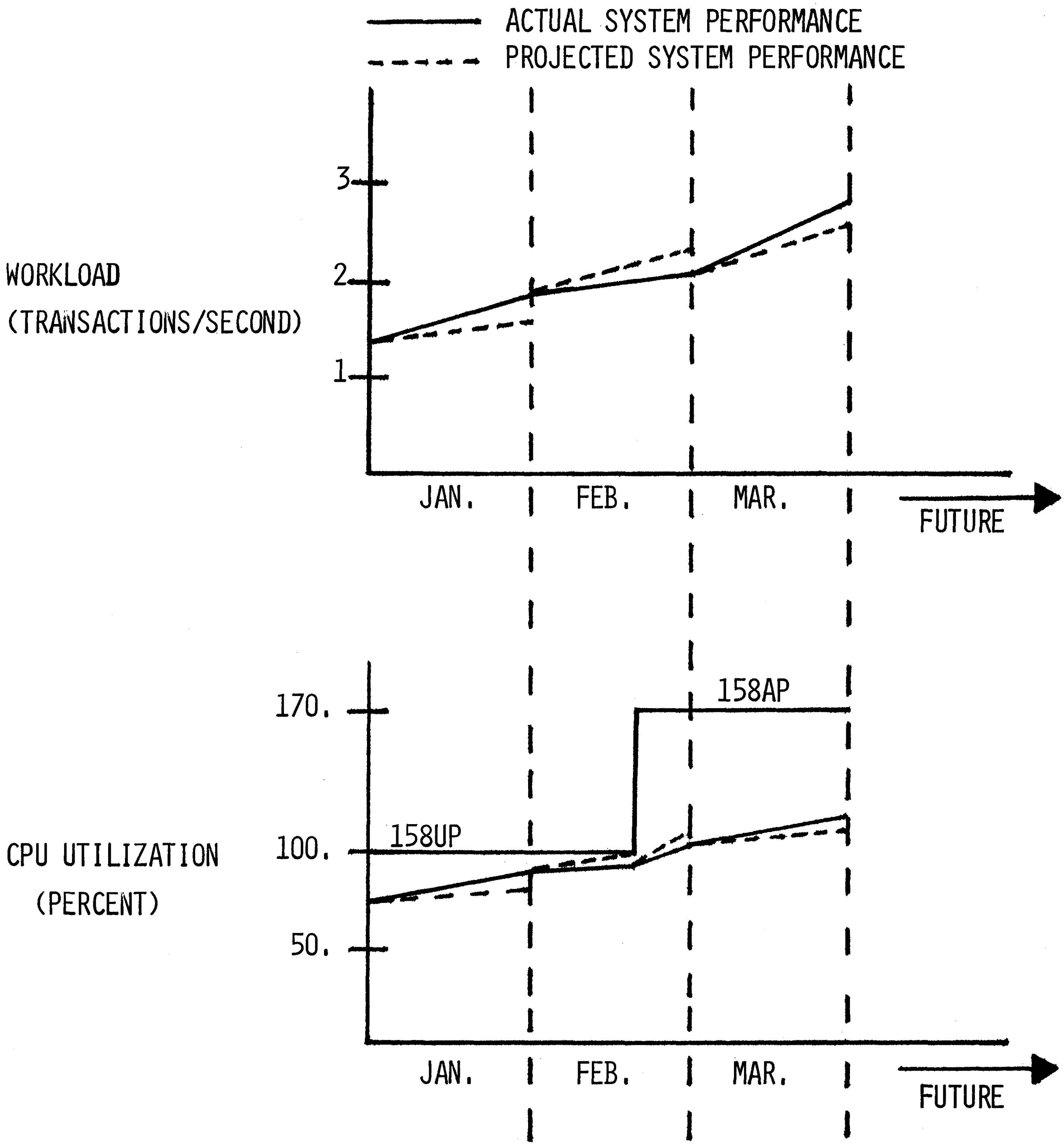
- Data collection
- Trend analysis and correlation
- New workloads and projects outlined
- Gross system requirements forecasted
- Data collection.

allows forecasts to be validated and other enhancements incorporated as development dictates. For example, there are certain measurement tool deficiencies. It may be reasonable at some later time to select another tool or to enhance the current measurement tools to obtain some additional parameters. Also, tracking and analysis begins to build confidence in various measured parameters. This means simple analytical techniques (e.g., single server queuing models) might enhance predicting capabilities. At this point, other automated models may provide certain insight into the system analysis. The point being that there is no real need to seek out complex modelling techniques until a degree of confidence in the data is obtained.

Summary

Capacity planning as a methodology must be viewed foremost as a vehicle to aid data processing installations in understanding their current environments. Its major role is to develop a structure in which the observed performance indicators may be related and thereby begin to understand the DP environment by experience and use the past to predict the future. For example, Figure 3 graphically depicts a computer facility where they have plotted

- The projected workload and processor utilization against the actual
- The total processor capacity for their 158 UP and 158 AP system
- The available processor capacity for each of these systems.



NOTE: CPU UTILIZATION PERCENTAGE IS BASED UPON CAPACITY OF 158UP.

FIGURE 3. SYSTEM PERFORMANCE.

When actual measurements are significantly different than the projections, these will require investigation. The investigations will improve the ability to make accurate future projections.

1.2 Basic Functions of Capacity Planning

In developing a methodology for capacity planning, it is important to understand why capacity planning is needed to effectively manage a computer installation. There are many computer operations where current resource utilizations are unknown and available capacity not understood (see Section 1.5). Also, the service being provided to users has not been quantified. For example, poor response time must be measured other than by unsatisfied users. More specifically, user on-line response time, batch turnaround time as well as other service type indicators must be quantified. A basic function of capacity planning is to quantitatively establish system performance levels (e.g., workload, utilization, user service) and track these on a continuing basis.

Scheduling provides a mechanism for exercising control over system performance and is an integral part of capacity planning. The system workload can be balanced through scheduling. Also, scheduling provides for the best resource utilization while attaining the required user service objectives.

Predicting performance, as a result of system changes (workload, hardware, software), is a significant part of managing a computer facility. The types of performance data (e.g., workload, service, response, etc.) required to develop and validate predictive models is an output of the capacity planning process.

A model and its predictions are assumed to be made for the tuned system, therefore, it is imperative that system bottlenecks be identified and removed when possible. A model developed in the face of certain resource bottlenecks does not represent the best possible system performance, hence, it will provide conservative estimates. Whereas, model projections made from a tuned system and at some later stage where tuning is allowed to seriously degrade (bottlenecking allowed to go unchecked), these projections will appear to be overly optimistic.

The kinds of performance data provided by capacity planning will also aid in system conversion efforts.

1.3 Personnel Requirements

The primary people involved in capacity planning are as follows:

- Users
- Systems Design and Programming Group
- Schedulers
- Capacity Planners
- Data Processing Managers
- Upper Management
- Operators.

As a means for discussing the interactions of the people involved in capacity planning as well as providing an overview of the process, a flow diagram (Figure 4) has been developed. The structural flow, indicated by the arrows, has no beginning or no end. This is as it should be, since the capacity planning process has been defined and developed to be a cyclic on-going management process. To best understand this structure, movement through the diagram should begin at the data processing scheduler. As shown, users contribute directly to two components of workload (current and new) imposed upon the scheduler. Users are those people involved in using the computer to do productive work. The Systems Design and Programming Group, as the name implies, is concerned with the design and programming of new applications. Also, if installations have more than one computing system installed, a workload shifted from a malfunctioning system would require scheduling on another system. Hence, it is incumbent upon the scheduler to accept this workload and schedule it upon the functioning systems.

For "optimum" operation, the scheduler may perform any possible load balancing. The computing system, composed of hardware and software, would have the necessary performance monitors for data gathering. Monitors are used to track service objectives and other performance indicators. Additional performance tools are made available to reduce and report upon system performance. Reports are output to DP Management, Capacity Planners, Users, Systems Design and Programming Group and Operations. The functions of most of

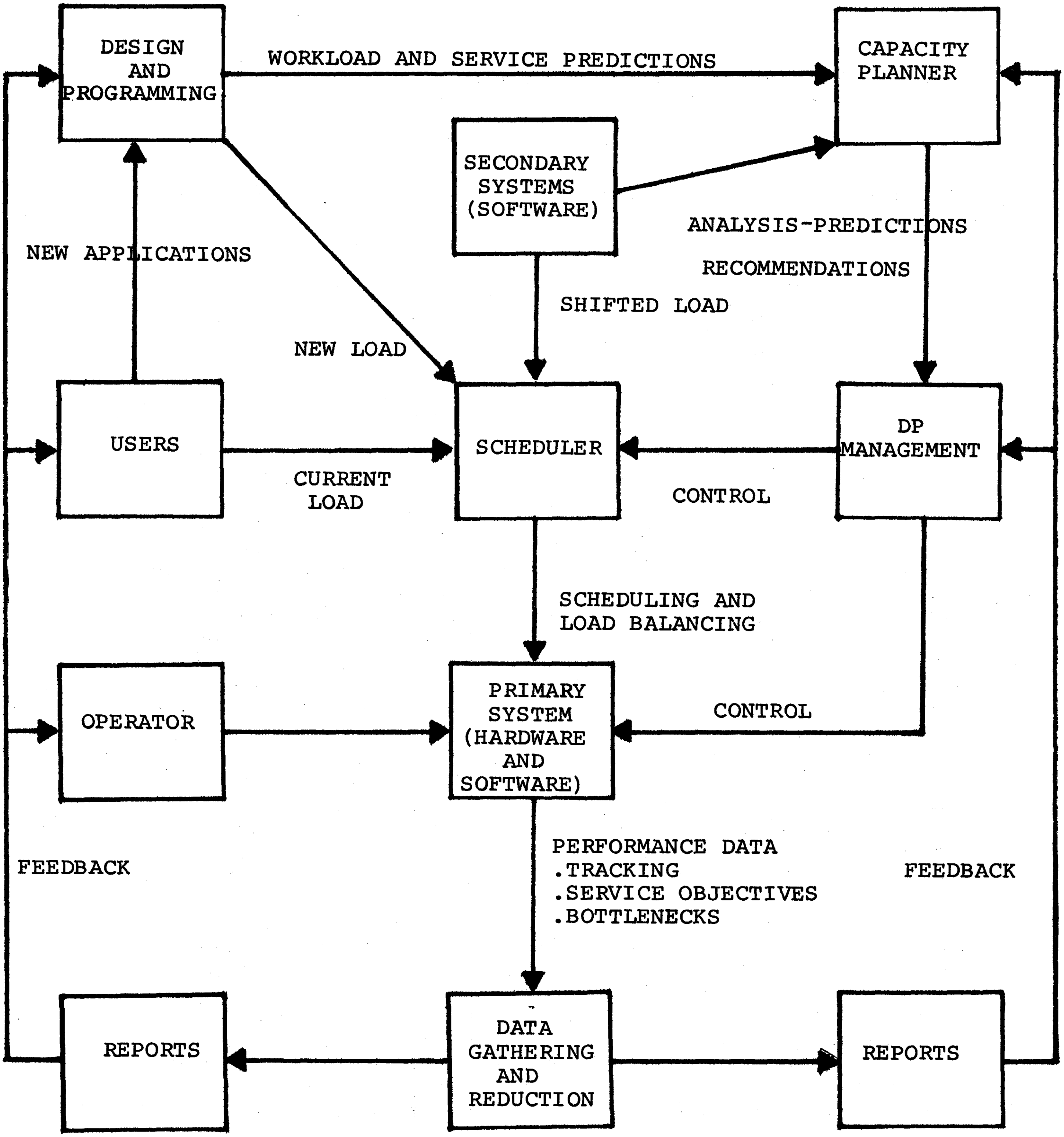


FIGURE 4. CAPACITY PLANNING FLOW.

these recipients are self explanatory except for the Capacity Planners.

Capacity Planners are concerned with system analysis, modelling and prediction. They are expected to analyze and draw conclusions from performance data collected by various software and hardware measurement tools. Therefore, the capacity planning group must have people knowledgable in the measurement tools area. System analysis will include the application software (IMS, CICS, TSO, etc.), the hardware (CPU, channels, control units, etc.) as well as the operating systems (MVS, SVS, VS1). Hence, systems analysts and operations personnel will become an integral part of the capacity planning group. An analytically oriented person (possibly from an operations research department) could enhance the group to provide modelling and predictive skills. As pointed out in the introduction, it is very critical that a person on the capacity planning team be designated as systems application co-ordinator. The capacity planning functions provided by the planners will require close coordination between all departments.

Planning for future hardware is based, in many instances, on increasing or decreasing loads on existing systems. But, a difficult problem is planning for loads created by new applications. It is hoped, as shown in Figure 4, that the Design and Programming Group will be able to make reasonable loading and service predictions for new applications. Therefore, from a practical point of view, gross estimation followed by measurement and validation appears to be the best approach.

Gross estimations may be made at two levels of new application definition. At the first level, the application may be designed in sufficient detail that all data base and data communication calls have been outlined for the different transaction types (on-line applications). In this case, path length data may be obtained for various functional activities within an access method. Then, knowing the approximate instruction execution rate of a given CPU and this path length information, gross estimations of CPU time per function can be calculated. The number of times a given function is called by a transaction and known transaction arrival rates, allows CPU time to be calculated. This time can be summed over various transaction types and the CPU time can be obtained. Also,

channel service time and I/O device activity can be calculated. This method allows new system resource requirements to be approximated. The one major service estimate that has been omitted is the workload service requirement imposed by the user written application code. Since projections made for new coding loads is difficult, the initial projection may be inaccurate. But, the feedback report (Figure 4) on new application performance will improve these projections. The critical factor in making such projections accurate is the feedback and comparison.

Estimations can be made at a second level, where the new application is not defined in detail, users have resorted to seeking out applications already installed which approximate their proposed application and projections are made based on performance data from the comparable applications. This method of planning for a new system may or may not be satisfactory. But, as stated earlier, the critical part of the process is measurement and comparison of the projection once the new application is installed. If this process is accomplished, assessment of the method is possible. Otherwise, it is never known whether such an approach is satisfactory. In many cases, this validation process is never carried out.

1.4 System Components

It was too large a task to bring together a working methodology for capacity planning which includes all operating systems and all possible subsystem. The concepts and approach do not change for different operating systems or subsystems; only their particular characteristics as they relate to system performance. The systems and subsystems being addressed are outlined below:

- Operating Systems
  - MVS
  - SVS
  - VS1
- Subsystems
  - TSO
  - APL
  - IMS
  - CICS
  - BATCH

Also, capacity planning addresses the management of the following hardware resources:

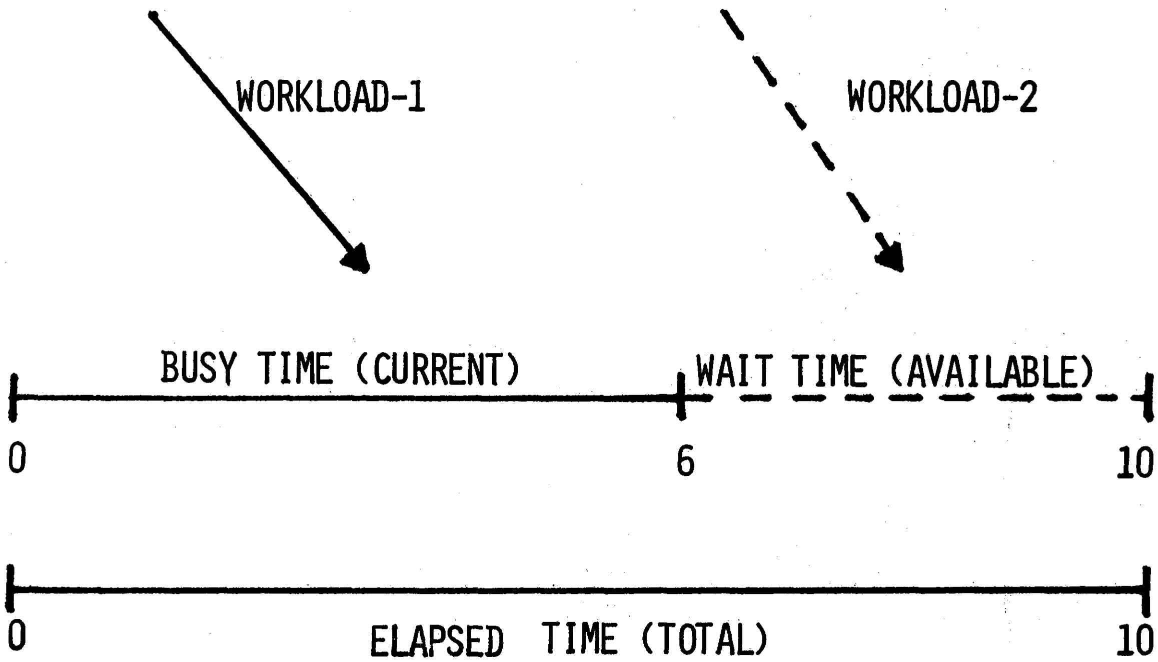
- CPU (UP, AP, MP)
- Channels
- Control Units
- DASD/Tapes
- Printers
- Communication Controllers
- Lines
- Cluster Controllers
- Controller Adapters
- Auxiliary Storage
- Terminals

This document is limited to the software and hardware components listed above.

1.5 Resource Capacity

In analyzing the performance of any hardware resource (CPU, channel, etc.), its total capacity may be broken into two components (busy and wait, Figure 5). The busy component indicates that portion of the resource currently being consumed. Current capacity is directly relately to the applied workload. Available capacity, in the strictest sense, is that portion not being consumed (wait time). Total capacity, is the sum of the two and defines the period of time over which the resource was available for use. From this definition of terms, utilization is busy time divided by the elapsed time.

In understanding capacity of resources, the two overiding factors of concern are available capacity and workload. Available capacity is very dependent upon workload. Also, there is a dependence on the resources required by the workload and how they interact. For example, consider a computing environment (Figure 6) where you have MVS, block multiplexor channels, and rotational positional sensing (RPS) devices. It is possible that a channel, although not saturated, will so impact the I/O response time at a channel utilization of 35% that transactions will experience excessive wait time. Assuming, for the case shown in Figure 5, that the resource being analyzed is the CPU, the CPU has an available capacity of 40 percent. This means 40 percent of the CPU cycles are available to do additional work. But,



$$\text{UTILIZATION} = \frac{\text{BUSY TIME}}{\text{ELAPSED TIME}} = \text{60 PERCENT}$$

ELEMENTS OF CAPACITY

- O TIME
  - o BUSY
  - o WAIT
  - o ELAPSED
- O UTILIZATION
- O WORKLOAD

FIGURE 5. RESOURCE CAPACITY.

AVAILABLE CAPACITY IS VERY DEPENDENT UPON WORKLOAD, IT'S REQUIRED RESOURCES AND THEIR INTERACTION.

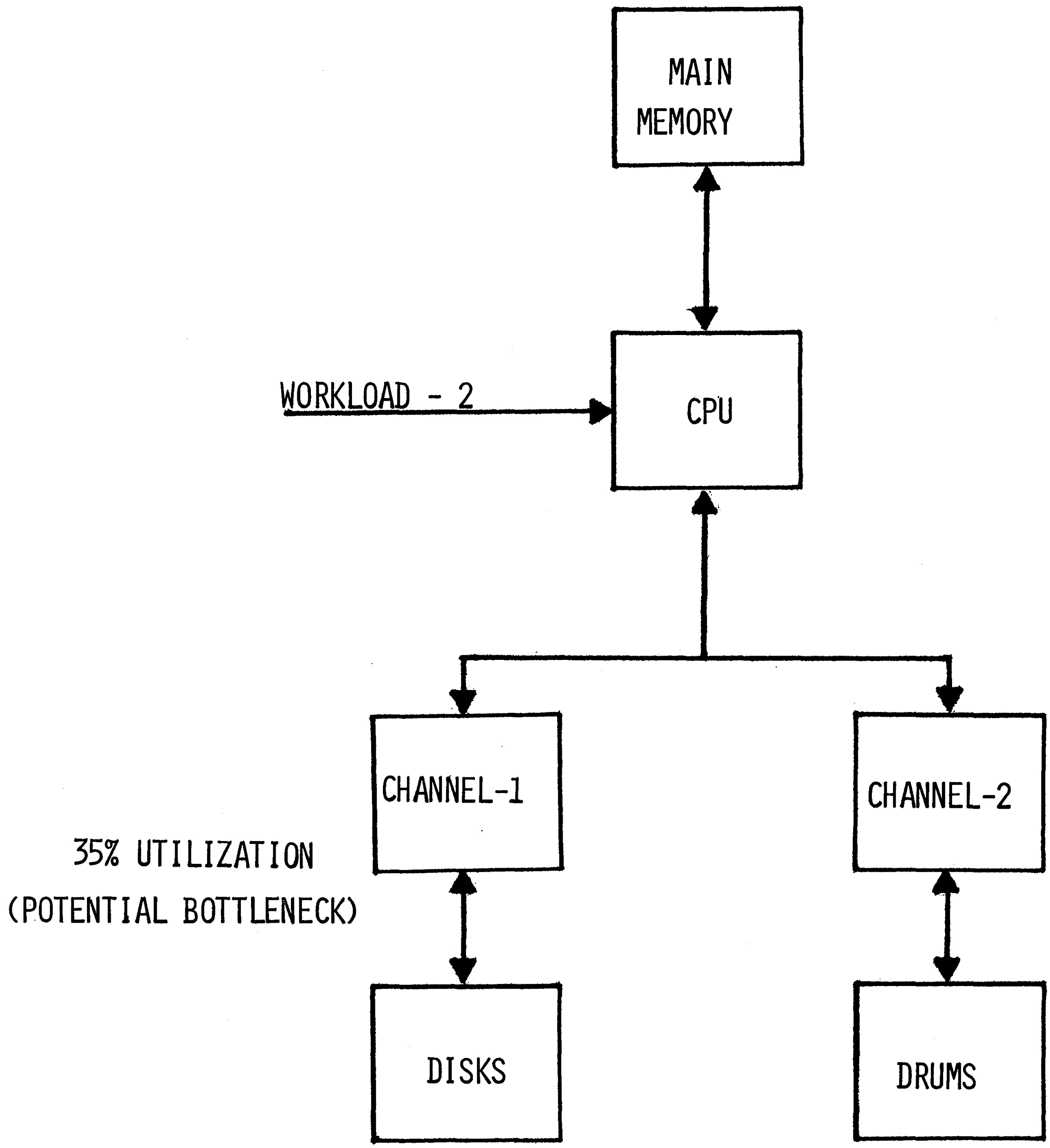


FIGURE 6. AVAILABLE RESOURCE CAPACITY.

if the workload shown in Figure 6 must access data on the bottlenecking resources (channel, RPS devices), it will spend an appreciable amount of time waiting for I/O and could use very little of the 40 percent available cycles. However, if the workload is a scientific job and basically CPU bound, it may use the entire available capacity. Hence, available capacity can only be understood by understanding the work to be accomplished and its system resource requirements.

1.6 System Capacity

System capacity, which is a function of many resources requires more than just an assessment of resource performance. To understand what is meant by system capacity, a computer room and all of the enclosed hardware and software may be viewed as a "black box". This black box has the function of providing a certain amount of service to a given user community. The system capacity or the capacity of this black box has been reached when any further increase in system workload causes certain critical user defined service objectives to be exceeded. This means, all the usual tuning, software updates or scheduling adjustments have been made and no further system improvements are possible. In the light of this simple definition, the critical elements which characterize system capacity are:

- User service objectives
- Workload
- Resource utilization.

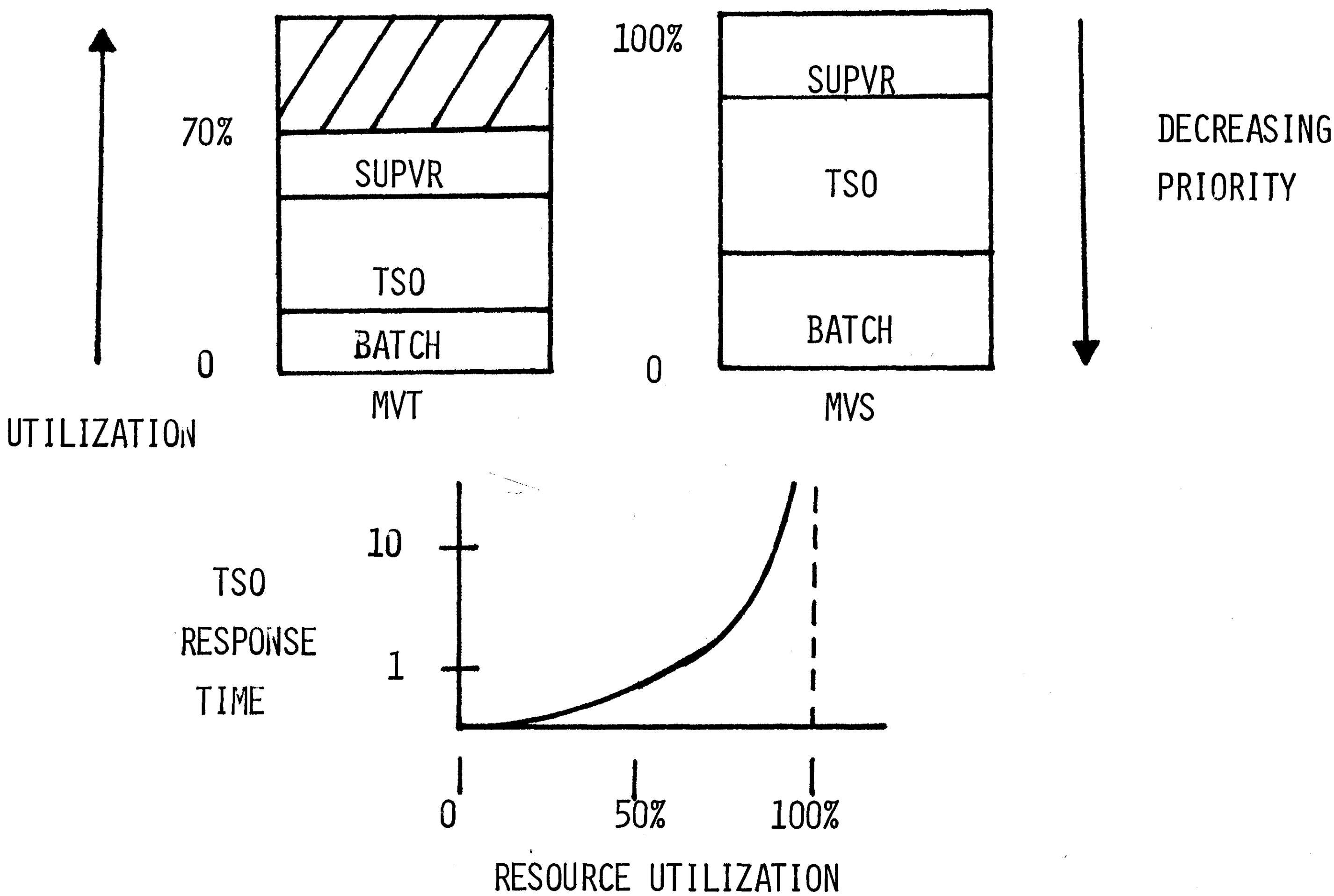
As pointed out, the most critical element in the assessment of system capacity is user service requirements.

To address the problem being encountered in some MVS installations, where the CPU is running for extended periods of time (6-8 hours) at 100 percent utilization, users are inputting additional work and the system continues to accept and process the increased load. The question is, "What are the capacity limits in this case (100% CPU utilization)?" Many customers were told, because of certain MVT bottlenecks (Job Queue/SVC LIB on DASD), CPU capacity limits were reached at average utilization values of 70 or 75 per cent (Figure 7). The major point to be understood from these factors are that system capacity cannot be quantified by CPU utilization alone.

O ELEMENTS OF CAPACITY

- o WORKLOAD
  - o BATCH (INITIATORS)
  - o ON-LINE (TERMINALS)
- o RESOURCE UTILIZATION
- o <u>USER SERVICE</u>*
  - o RESPONSE TIME
  - o TURNAROUND TIME

O CAPACITY CONSIDERATIONS (CPU)



* MOST CRITICAL INDICATOR OF SYSTEM CAPACITY

FIGURE 7. SYSTEM CAPACITY

In a prioritized system, as shown in Figure 7, workload levels can be traded-off (TSO vs BATCH) in either case (MVT or MVS). This means, although, the CPU appears to be out of capacity, the system may or may not be out. For example, in either system, the number of TSO terminals may be increased (increased workload) at the expense of the batch environment (elogation of batch job turnaround time). As long as the degredation in TSO response time or elogation in batch turnaround time does not exceed certain predefined objectives, system capacity has not been exceeded. But, the level of resource utilization by each subsystem is an indicator of the rate at which user service objectives will degrade for a given increase in workload. For example, as shown on the graph in Figure 7, the TSO response time degrades faster as more of the CPU utilization is due to TSO work. Such that, as 100 percent CPU utilization is reached (all TSO workload), a small increase in workload will be manifest in a sharp increase in TSO response time.

Another factor in managing system capacity is the establishment of reasonable service objectives for various environments. There are certain trade-offs to be considered in such an assessment. In providing users a certain response time in a TSO or IMS environment, a trade-off can be made between the number of users and response time. Obviously, each application must be analyzed and the best trade-off criteria established. For example, a case might be one of determining the number of terminals or clerks required for handling customer service inquiries in a utility company. A very fast response time would improve the productivity of the clerks, but, the CPU service requirements would be more stringent and costly. Therefore, clerk and terminal costs may be traded-off for CPU costs. Obviously, the customer workload (inquiries per second) will indicate a certain minimum number of terminals. Publications indicate what might be considered reasonable response and think times for certain applications. This kind of input, along with the users own in house investigations, should allow him to get a handle on service objectives for his installation. There would be corresponding trade-off issues in the batch environment.

In the process of quantifying system capacity, there is also the need to control the user service objectives. This arises from the fact that user workloads increase very rapidly when their service is drastically improved by

certain system improvements (e.g., purchase of another CPU). The CPU may have been purchased to provide capacity for another application but before the new application is put on-line, old application workloads have increased and consumed this new capacity. Had response and turnaround times been controlled and not allowed to improve with increased capacity, then, most likely old application workloads would have grown at "normal" rates. Probably some type of service governor might be employed if appropriate (e.g., build in response time delay).

1.7 Unattainable Capacity

When a computer system is installed, it provides a certain amount of total capacity. In a working environment, where it is possible to run each resource at 100 percent utilization and the system is 100 percent available, total system resource capacity would be 100 percent. The point is, that the probability of attaining 100 percent total system capacity is very small because there are certain components of total capacity which are unattainable. Unattainable in the sense that theoretically it is available but in a practical sense it is unreachable. Some of the factors that tend to reduce total system capacity are described below.

- System Down Time

  Capacity lost due to system down time may be attributed to preventive maintenance or system repair due to malfunctioning hardware or software. Also, work in process at the time of failure may have to be rerun. Time is also lost to restart in many situations (15 to 30 minutes).

- Operational Problems

  There are many ways that capacity may be lost on the operations floor and the following are two examples. Operator changes at the close of a shift. In one environment as much as one-half hour was lost because the operators were not ready to begin at the end of the preceding shift. Job rerun time due to tapes being mounted on incorrect drives.

- Program and Data Problems

  Problems that arise with the program or its input data will cause a loss of capacity due to reruns. For example, when tape drives are not cleaned frequently, tape errors may require jobs to be rerun.

- Variations in Scheduling Demands

  The input work does not flow into the computer in a regular fashion, therefore, the computer may not be utilized 100 percent during all periods (e.g., lunch breaks, weekends, etc.). Changing schedules due to higher priority work is ever present in a computing environment. This means schedules may be established for a 24 hour period but introduction of priority jobs will push scheduled work further out in time.

- System Recovery

  System recovery, means reruns are particularly cumbersome if the rerun is a predecessor-feeder program. This impacts capacity by requiring the jobs dependent upon the predecessor jobs to be rescheduled. This condition exists even though the system has capacity available at the dependent jobs normal start time.

Chase Manhattan Bank has collected values for various components of unattainable capacity and their values (Figure 8) show that unattainable capacity is a real factor to be considered in capacity planning. This example is for the MVT operating system. The percentages are part of a factor termed "Relevant Capacity" by Chase Manhattan Bank and applies only to the CPU resource. The Relevant Capacity of a CPU is that capacity remaining after the capacity required by the operating system has been removed. To summarize the figure, Chase Manhattan Bank indicates that 36 percent of their Relevant Capacity is unattainable and 64 percent is available for scheduling their required workload. The reason for addressing these factors of unattainable capacity is to establish that portion of the overall capacity available for scheduling current as well as future work. Obviously, the useful life of a computer system cannot be predicted if the maximum capacity is not properly understood and established.

Example: Reference Computerworld article by Chase Manhattan Bank entitled "100% Utilization, Impossible Dream", dated February 19, 1975.

Factors acting to reduce effective CPU capacity:

1. Operational ineffectiveness	9%
a. System downtime	
b. Operational problems	
c. Program and data problems	
2. Unreachable capacity	15%
a. Difference in system requirements	
3. Recovery	5%
a. Predecessor-feeder relationships	
4. Variations	7%
a. Operational ineffectiveness	
b. Arrival of scheduled demand	
Total	36%
Threshold capacity 100 - 36 =	64%

Figure 8. Unattainable Capacity.

2.0 Performance Measurement

2.1 Introduction

The primary capacity planning measurement requirements are for loading and service data. Although parameters indicating saturation and bottlenecking within a computing system are of definite concern. From a workload standpoint, the concern is to measure the number of different job types arriving per hour in a batch environment or transaction types arriving per second in an on-line environment (Figure 9). In most instances, measurement tools (SMF, CICS Performance Analyzer, IMS/VS Monitor Report Print Program, etc.) available today are able to do a satisfactory job at measuring system workload.

When the workunit has been defined, the subsystem (software and hardware) service requirements by workunit type must be measured. In this area, accuracy and repeatability are very important.

To adequately characterize service requirements, it is necessary to have hardware and software monitors (see Section 2.3). As shown in Figure 9, a transaction or job will pass through several hardware subsystems in completing its processing cycle. At each subsystem, the service time (busy time - utilization) for a job or transaction type is to be measured. In most instances, the service time component added by the CPU is complex and more difficult to measure. As shown in Figure 10, CPU service time should be broken down on a module by module basis. This allows the analyst to see for a given job or transaction type where the primary processing time is spent. As for accuracy and repeatability, deviations in measurements may be analyzed to see those modules that show large variations. Then these variations may be traced back to updates (e.g., new releases, selectable units) or certain module inconsistencies. In trying to understand service as provided in a batch, TSO, CICS, IMS, etc. environment, service data would be reported as shown under the "Required Data" section of Figure 10. This is an attempt to move away from cataloging CPU time into problem program and supervisor state. Trying to develop a capacity planning methodology that uses supervisor and problem program CPU time has led to confusion in the way various measurement tools view these states. As outlined in Figure 10, each job or transaction

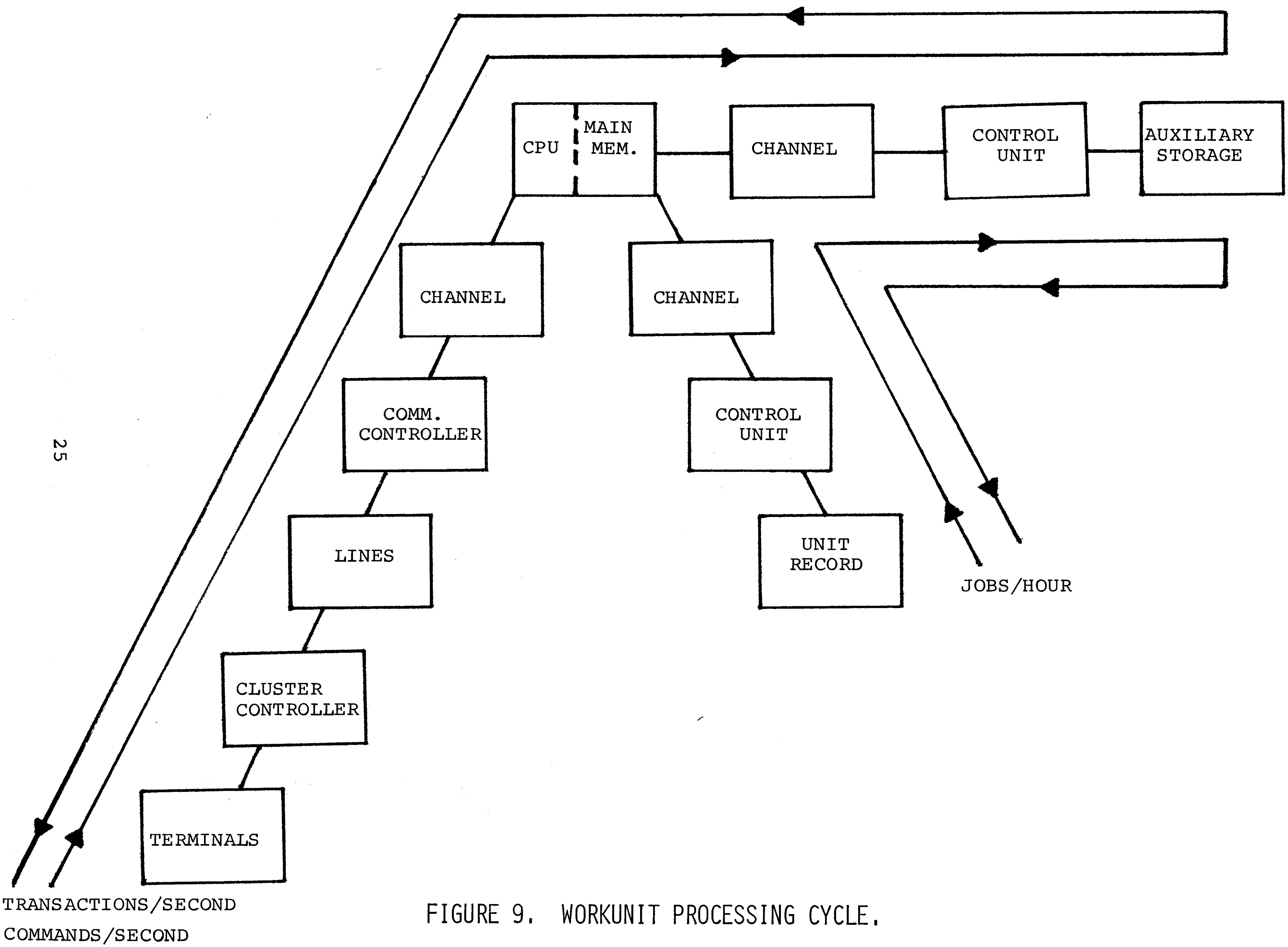
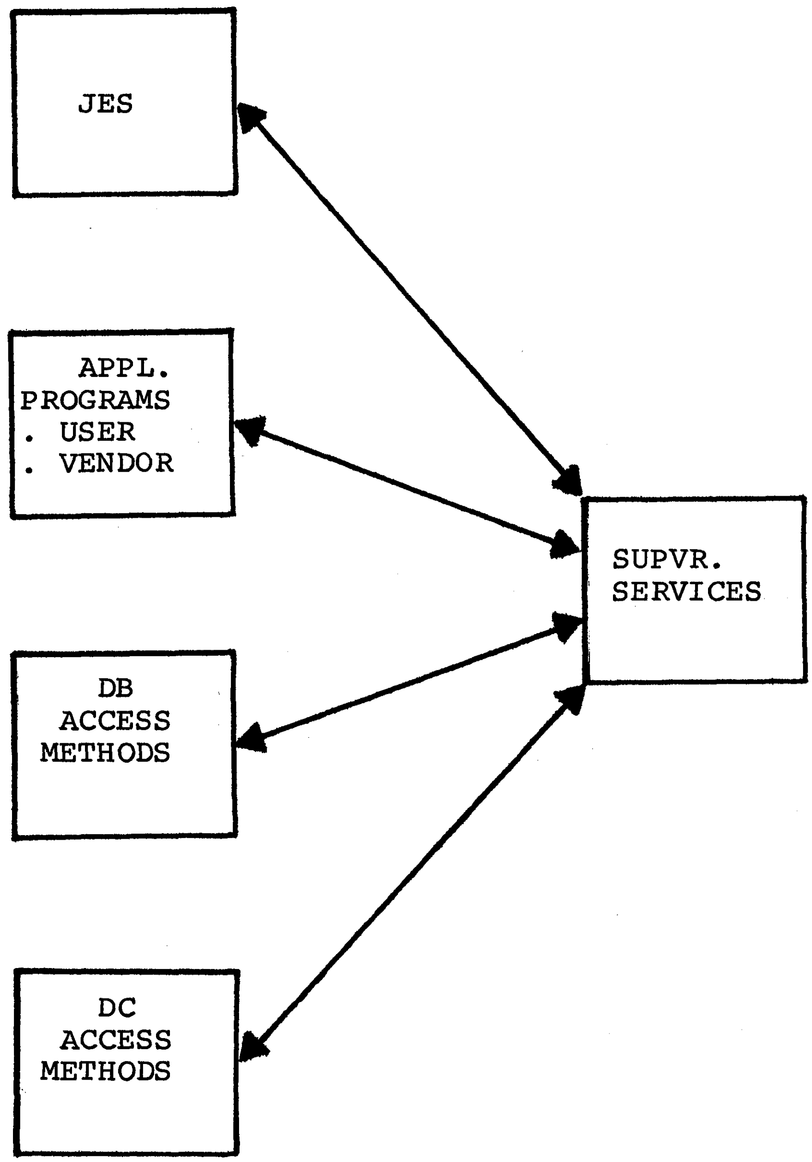


FIGURE 9. WORKUNIT PROCESSING CYCLE.



REQUIRED DATA

JOB-TRANSACTION-COMMAND	TYPE	MODULE	CPU TIME
JOB-A		JES	XX
		APPLICATION	XXXX
		DB ACCESS METHOD	XXX
		SUPERVISOR SERVICES	XXX
EDIT		APPLICATION	XXX
		DB ACCESS METHOD	XXX
		SUPERVISOR SERVICES	XXX

FIGURE 10. WORKUNIT CPU TIME.

is accountable for any service time provided to it by any module. For example, the dispatching time required of the supervisor to dispatch a task for transaction Type-A would be added to its supervisor services component (Figure 10).

In most instances, the tools available for measuring CPU time do not break it out on a module by module basis. Much of the confusion in trying to interpret measured output data comes from the fact that certain modules are omitted all together or supervisor and application CPU time are summed together and recorded as application time. Also, this time might include some access method time. In cases where module times are not delineated, measurement times may change for a given job because of an increased supervisor load (i.e., longer chains to search). Hardware and software monitors may disagree in measuring a particular parameter because of a non-uniform approach to measuring. For example, a software monitor will record the EXCP load on channel and with the number of BYTES/EXCP we can calculate channel utilization based on the device data transfer rate. This utilization in most instances will not correlate with the same value recorded by the hardware monitor because the hardware monitor measures the total busy time. This will include the time for control characters as well as data and depending on block sizes these additional control characters can be significant. Therefore, when channel utilization is being analyzed from an EXCP point of view be aware of the inaccuracies. A solution to this problem might be to use a software monitor which samples channel busy (e.g., VS1PT, SVSPT) or a hardware monitor and develop a factor to be used for channel overhead when channel utilization is to be calculated from an EXCP rate. One rule of thumb is to double data transfer time to obtain total channel busy time.

The format for reporting data is also a critical item in capacity planning. The number of reports should be kept to a minimum and reports selected must be easy to review. Most reports available today are displayed in long columns of numbers which make it very difficult to review and compare points in time. One major aid in this area would be to report data in a graphical format (Figure 11). Some graphical report writers are available for MVS, SVS, VS1, CICS, TSO, IMS but these are limited. As shown in Figure 11, it is very easy from a graphical presentation to compare a point at 6:00 AM with another at 12.00 PM. Currently many users are producing their own graphical report writers.

CPU UTILIZATION VS TIME

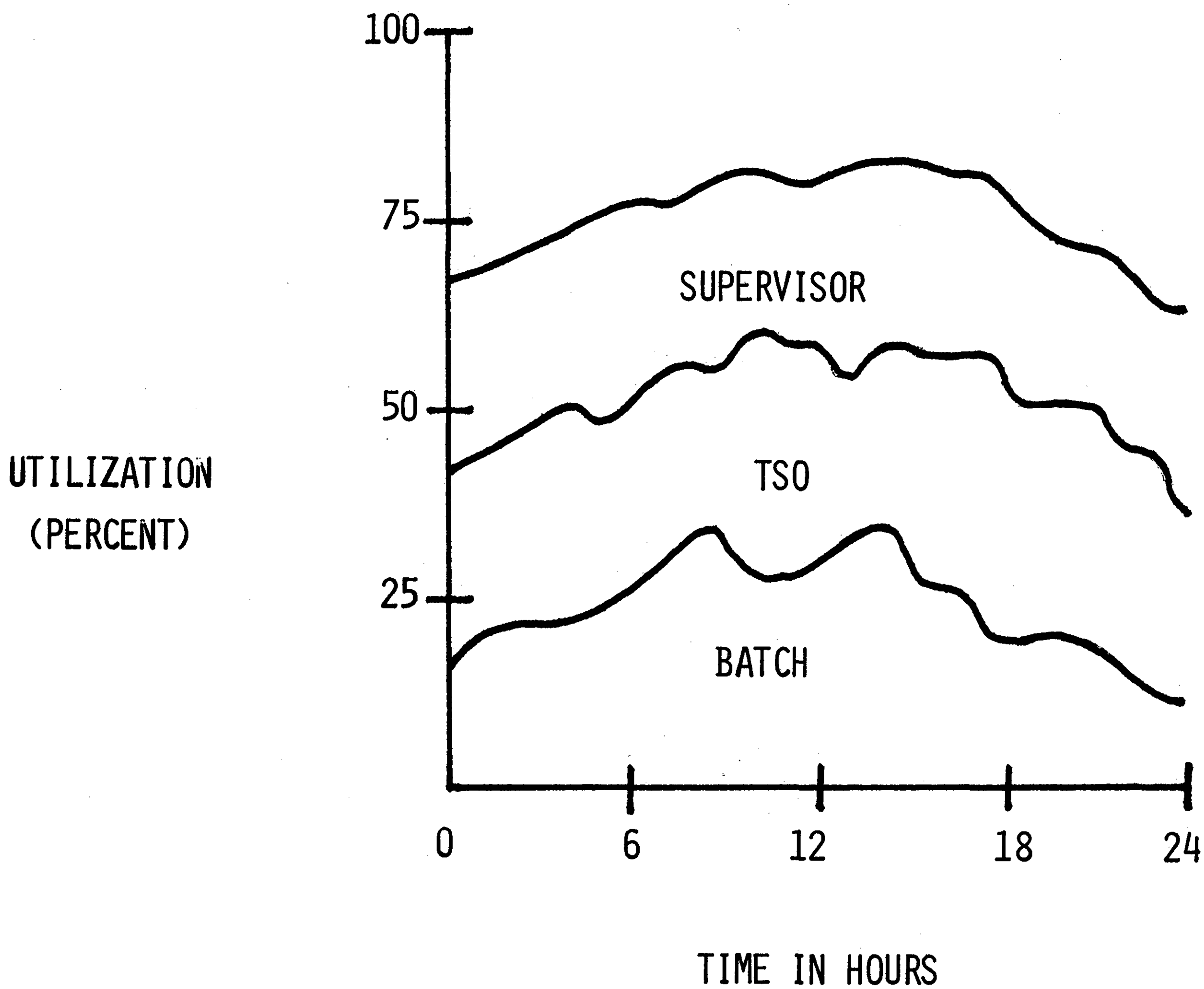


FIGURE 11. GRAPHICAL DISPLAY.

2.2 Measurement Methods

Most tools employ some type of sampling technique in the measurement of systems. They may sample counters which summarize certain system activity or sample system status at prescribed intervals. Sampling may also be controlled by certain system events (e.g., page fault, I/O interrupt, etc.), where a module is activated to gather data upon the occurrance of each event.

Counter sampling requires relatively little overhead. Sampling may be done relatively infrequently and the volume of data produced is fairly small. In this method of measurement (counting), there is no sampling error. Measurement tools which currently employ this method are SMF, MF/1, RMF, VS1PT and SVSPT (partially).

In status sampling, a program gains control at specified intervals and records the instantaneous status of various system components. It is very important to note that this method is subject to sampling errors which increase with a decrease in sampling frequency. Measurements are generally distorted by an inability to gain control while higher priority tasks are running or while the system is disabled. The primary advantage of this method is that no overhead is incurred between samples. Measurement tools currently using this method are RMF, MF/1, VS1PT and SVSPT.

When sampling by event, the monitoring program gains control at the occurrance of some specified event. At this time system status data as well as counter contents may be recorded. In using this method, the greatest system overhead is usually incurred and large volumes of data produced. One tool using this method is the Generalized Trace Facility (GTF).

2.3 Measurement Tools

Measurement tools used in capacity planning are categorized as hardware or software monitors. As a further breakdown of the tools discussed in this section, the hardware and software monitors are categorized as collectors, analyzers, or collector-analyzer (Figure 12). Collectors are those tools that gather system data and records it on some permanent copy medium (i.e.,magnetic tape, disk or drum). In general, collectors do not perform any data reduction.

- Collectors

CA - Hardware monitor
C2 - SMF (system measurement facility)
C3 - GTF (general Trace Facility)
C4 - TS Trace (Time Sharing Trace)
C5 - IMS/VS System Log

- Analyzers

A1 - Hardware monitor report program
A2 - SGP (Statistics Generating Package)
A3 - SMF Graphical Analyzer
A4 - Capacity Management Aid
A5 - IMS/VS Log Transa tion Analysis
A6 - IMS/VS Statistical Analysis

- Collector - Analyzer

CA1 - MF/1 (Measurement Facility 1)
CA2 - RMF (Resource Measurement Facility)
CA3 - VS2PT (VS2 Performance Tool)
CA4 - VS1PT (VS1 Performance Tool)
CA5 - SIR (System Information Routine)
CA6 - CICS Performance Analyzer II
CA7 - CICS Plot
CA8 - CICS Dynamic Map
CA9 - IMS/VS Monitor Report Print Program
CA10 - IMS/TRAPDL1
CA11 - APL System
CA12 - Utility IEHLIST (List VTOC)

Figure 12. Performance Measurement Tools.

Analyzers use collector data as input. This data is reduced and a readable report produced. As shown on Figure 11, in most cases each collector has an associated analyzer. When this is not the case, users normally write their own specialized analyzer. The collector-analyzer is, as the name implies, a tool which collects, reduces and prints a readable report.

Since certain measurement tools apply only to a specific operating system, a listing (Figure 13) is provided that outlines each tool and the applicable system. Also, the tool availability is recorded in the type column. This availability is defined as given below:

- SCP - Provided with System Control Program (SCP)
- PP - Provided with Program Product (PP)
- FDP - Provided as a Field Develop Program (FDP)
- IUP - Provided as a Installed User Program (IUP).

2.4 PERFORMANCE DATA REQUIREMENTS

In this section many performance parameters are discussed. These parameters have been categorized by their usage (Figure 14) for planning purposes as well as to remove some of the complexity. Before capacity planning can proceed effectively, it is necessary to establish the current state (performance level) of the system. By this, it is meant that current workloads, resource utilizations, response times, etc. must be well defined. Also, there is a certain subset of the performance data used to determine the system tuning level. The question is always asked, "What is considered a well tuned system?" It is not the intent of this bulletin to address tuning in detail or to answer this question but to note that a reasonably "well tuned" system is required to initiate a capacity planning effort. The results from the assessment of the current system reflects upon all future predictions and forecasts.

Since performance tracking is a continuing effort. It is very important to minimize tracking data requirements. The integrity of long range projection is a function of system tuning and must be tracked. Projections will not

		TYPE	MVS	SVS	VS1
1.	Hardware Monitor		X	X	X
2.	SMF	SCP	X	X	X
3.	GTF	SCP	X	X	X
4.	TS Trace	SCP	NA	X	NA
5.	IMS/VS System Log	PP	X	X	X
6.	Hardware Monitor Report Program		X	X	X
7.	SMF Graphical Analyzer	FDP	NA	X	X
8.	Capacity Management Aid	FDP	X	X	X
9.	SGP - Statistics Gathering Pkg.	FDP	X	X	X
10.	IMS/VS Log Transaction Analysis	PP	X	X	X
11.	IMS/VS Statistical Analysis	PP	X	X	X
12.	MF/1	SCP	X	NA	NA
13.	RMF	PP	X	NA	NA
14.	SVSPT	IUP	NA	X	NA
15.	VS1PT	IUP	NA	NA	X
16.	SIR - System Information Routine	IUP	X	NA	NA
17.	CICS Performance Analyzer II	FDP	X	X	X
18.	CICS Plot	FDP	X	X	X
19.	CICS Dynamic Map	FDP	X	X	X
20.	IMS/VS Monitor Report Print Program	PP	X	X	X
21.	IMS/TRAPDL1	FDP	X	X	X
22.	APL System	PP	X	X	X
23.	Utility IEHLIST (List VTOC)	PP	X	X	X
24.	MVS/SVS System and Job Input Analysis	IUP	X	X	NA

Figure 13. Use of Performance Measurement Tools.

- ESTABLISH CURRENT STATE OF SYSTEM
- ESTABLISH TUNING LEVEL
- MODEL DEVELOPMENT (PREDICTIONS/FORECASTS)
  - HAND CALCULATIONS
  - AUTOMATED
- MODEL VALIDATION
- PERFORMANCE TRACKING
  - TUNING
  - PROJECTIONS (PREDICTIONS/FORECASTS)
  - SCHEDULING

Figure 14. Performance Data Usage.

automatically happen but must be made to come true. This may be brought about by tracking projections and analyzing in detail those points where the actual begins to deviate from the forecasts. In this manner, the problem (poor projections, bottlenecking, wrong assumptions, etc.) associated with the deviation may be clearly defined and understood. Since turnaround times are so critical in the batch environments certain scheduling paraments must be tracked. The specific data to be recorded in each of these categories is outlined later in this section.

Prediction and forecasting requires the collection of performance data for model development. In many instances, all the data required for modeling is not available. Hence, modelling involves data approximations in many cases. To support the modelling techniques discussed in this bulletin, all the data requirements are outlined but data deficiencies are resolved as they arise in each analysis. After a given model has been developed, a subset of this data is gathered for the validation phase.

In the analysis of current performance as well as the assessment of future requirements certain data recorded is system wide. This data reflects the operating system being used (MVS, SVS, VS1). Figure 15 outlines the specific measurement tools for gathering this data and the applicable operating system. The system data to be measured is defined in matrix form by Figure 16 and 17. This data is categorized by resource CPU, main memory, channels and I/O devices. An "X" at a given intersection of a column (performance tool) and a row (performance parameter) of the matrix indicates the performance tool will collect the parameter directly. If an "(X)" is located at the intersection, the performance tool only partially collects the required parameter and further reduction is necessary.

To understand and characterize each type of work being performed on a computer system (i.e., batch, TSO, APL, CICS, IMS) several figures have been developed outlining the data required for this process. Also, this data may be used as input to queueing models as appropriate. Basically, each figure outlines subsystem workload (CPU and I/O) and service (CPU) requirements (CPU). The figures are organized in matrix format where each column is headed by a code. This code (see Figure 12) identifies a specific performance measurement tool. An "X" placed at the intersection of a

	MVS	SVS	VS1
Hardware Monitor	X	X	X
SMF	X	X	X
SGP	X	X	X
GTF	X	X	X
MF1 - RMF	X		
SVSPT		X	
VS1PT			X
SIR	X		
IMPACT ANALYZER	X	X	

Figure 15. Performance Measurement Tools for Operating Systems.

PARAMETERS \ MEASUREMENT TOOL	HDW. MON.	SMF	SGP	MF/1	RMF	SVSPT	VS1PT	SIR	GTF	IMPACT ANAL.			
CPU													
. TOTAL WAIT	X	X	X	X	X	X	X						
. IDLE WAIT	X	(X)	(X)			X							
. I/O WAIT	X					X	X						
. PAGE WAIT						(X)	X						
. UTILIZATION	X	X	X	X	X	X	X	(X)		X			
. CPU TIME (PROBLEM PROGRAM)	X	(X)	(X)					(X)					
. CPU TIME (SUPERVISOR)	X							(X)		X			
. SYSTEM PAGING RATE	(X)			X	X				X	X			
. USER PAGING RATE	(X)	(X)	(X)	X	X				X	X			
. TOTAL PAGING RATE	(X)			X	X	X	X	X	X	X			
. SWAPPING RATE		(X)	(X)	X	X	X							
. PAGES PER SWAP-OUT		(X)	(X)	X	X	(X)							
. PAGES PER SWAP-IN		(X)	(X)	X	X	(X)							
. CPU, CHANNEL OVERLAP	X			X	X	X	X						
. MULTIPROGRAMMING LEVEL BY TIME		(X)	(X)	(X)	(X)	X							
. NUMBER ACTIVE INITIATORS BY TIME		(X)	(X)			X	X	X					
MEMORY													
. AVAILABLE FRAMES BY TIME				X	X	X	X	X		(X)			
. WORKING SET SIZE BY USER		(X)	(X)			(X)	X	X		X			

FIGURE 16. SYSTEM PERFORMANCE PARAMETERS (PART I).

PARAMETERS \ MEASUREMENT TOOL	HDW. MON.	SMF	SGP	MF/1	RMF	SVSPT	VS1PT	SIR	GTF	IMPACT ANAL.			
CHANNEL													
. UTILIZATION	X			X	X	X	X						
. TOTAL EXCP'S	X	(X)	(X)	(X)	(X)	(X)			X				
. BYTES/CHANNEL/TIME	X								X				
. CHANNEL QUEUE SIZE (RQE)						X							
I/O DEVICE													
. UTILIZATION				X	X	(X)	X						
. EXCP'S/DEVICE	X	(X)	(X)	(X)	(X)	X			X				
. EXCP'S/DEVICE/JOB									X				
. BYTES/DEVICE/TIME	X								X				
. DATA SET ACCESS RATE/JOB		X	X										
. DEVICE QUEUE SIZE				X	X	(X)							
. AVERAGE BYTES/EXCP	X												

FIGURE 17. SYSTEM PERFORMANCE PARAMETERS (PART II).

column and row (performance parameter) indicates the tool which collects the performance parameter.

The data required for capacity planning in the batch environment is outlined in Figure 18. Loading data is grouped in two categories. First, there may be so many batch jobs that it is not practical to analyze each one individually. Hence, jobs may be grouped by some distinguishing characteristics (e.g., required resources, approximate run time, approximate CPU time, etc.), then, as the first category indicates, it is possible to record the average number of jobs arriving per hour per group at the CPU. This workload generates an average I/O load as indicated on the figure.

The second category of batch work is large production jobs that consume large amounts of system resources. For this reason, they should be analyzed individually and not as part of some larger group. As indicated on the figure, the data gathered is the same as that obtained for groups.

Service data indicates the CPU processing time required on the average to execute a job arriving for a particular group. Also, the job elapsed time and group throughput is recorded. As indicated under loading data, service requirements for large product jobs are recorded separately. Use of main memory by working set is also collected.

TSO capacity planning data (Figure 19) is categorized in the same way as previously discussed for the batch environment. In the TSO environment, the workload is characterized as commands per second. When there are many command types, they are usually identified by classes. There are several other parameters included under the loading category which are self explanatory.

The TSO service requirements are recorded as the CPU processing time required on the average to process a command in a given class. Also, the amount of CPU time required on the average to perform a swap (in and out) is recorded. There are several other parameters listed under service but these should be self explanatory.

The similarity between APL capacity planning data (Figure 20) and TSO (Figure 19) is very evident. Therefore, no real discussion is required for this figure except to note that

MEASUREMENT TOOL	C2	A2	A3	A4	CA5	CA12
MEASURED DATA						
LOADING						
• BY GROUPS TRACKED BY INSTALLATION						
• JOBS ARRIVING/HOUR	X	X				
• EXCP's/CHANNEL	X	X				
• EXCP's/DEVICE	X	X				
• BYTES/EXCP						X
• BY LARGE PRODUCTION JOBS						
• EXCP's/CHANNEL	X	X				
• EXCP's/DEVICE	X	X				
• BYTES/EXCP	X	X				X
SERVICE						
• BY GROUPS TRACKED BY INSTALLATION						
• CPU TIME/JOB	X	X				
• ELAPSED TIME/JOB	X	X				
• JOBS COMPLETING/HOUR	X	X				
• BY HEAVY PRODUCTION JOBS						
• CPU TIME	X	X				
• ELAPSED TIME	X	X				
RESOURCE UTILIZATION						
• AVERAGE MEMORY WS SIZE BY CLASS						X
• AVERAGE MEMORY WS SIZE BY LARGE JOB						X
• GRAPHICAL ANALYSIS				X	X	

Working Set (WS)

Note:
- C2 - SMF
- A2 - SGP
- A3 - SMF Graphical Analyzer
- A4 - Capacity Management Aid
- CA5 - SIR
- CA12 - Utility IEHLIST

Figure 18. Batch Capacity Planning.

MEASURED DATA / MEASUREMENT TOOLS	C2	C3	C4	A2	CA1	CA2	CA5
LOADING							
• TRANSACTIONS/SECOND/CLASS		X	X		X	X	
• MEAN NUMBER CONCURRENT USERS/PERIOD	X			X			
• MEAN NUMBER SWAPS/TRANSACTION			X		X	X	
• AVERAGE PAGING LOAD/SWAP			X		X	X	
• TOTAL EXCP'S/DEVICE/PERIOD	X			X			
• TERMINAL I/O LOAD BY USER	X			X			
• CONNECT TIME	X			X			
SERVICE							
• AVERAGE CPU TIME/CLASS		X	X		X	X	
• TOTAL CPU TIME/PERIOD		X	X		X	X	
• TOTAL ELAPSED TIME PERIOD		X	X				
• AVERAGE USER RESPONSE TIME		X	X				
• AVERAGE THINK TIME		X	X				
• AVERAGE CPU TIME/SWAP		X	X				
RESOURCE UTILIZATION							
• AVERAGE MEMORY WS SIZE BY USER							X
• AVERAGE MEMORY WS SIZE (TOTAL)		X	X				

Note:
C2 - SMF
C3 - GTF
C4 - TS Trace
A2 - SUP
CA1 - MF1
CA2 - RMF
CA5 - SIR

Figure 19. TSO Capacity Planning.

MEASURED DATA	MEASUREMENT TOOL CA11	C2	A2	CA5
LOADING				
• TRANSACTIONS, SECOND/CLASS	X			
• MEAN NUMBER CONCURRENT USERS/PERIOD	X			
• MEAN NUMBER SWAPS/TRANSACTIONS	X			
• TOTAL EXCP'S/DEVICE/PERIOD		X	X	
SERVICE				
• AVERAGE CPU TIME/CLASS	X			
• AVERAGE CPU TIME/SWAP (IN & OUT)	X			
• AVERAGE THINK TIME	X			
• TOTAL CPU TIME	X			
• AVERAGE USER RESPONSE TIME	X			
RESOURCE UTILIZATION				
• TOTAL WORKING SET SIZE				X
• WORK SPACE WS SIZE AT SWAP	X			

Note:
CA11 - APL System
C2 - SMF
A2 - SGP
CA5 - SIR

Figure 20. APL Capacity Planning.

the average work space working set size is collected under resource utilization.

For the two TP application packages CICS and IMS, the data requirements are outlined in Figures 21 and 22. The basic requirements are for loading and service data. It should be noted for CICS file loading data, the access method being used must be identified. The CPU time required to process I/O file activity is very dependent on the access method. As outlined, many data points are required. In most instances, all points are not available which means in certain cases approximations must be made. Also, analysis can proceed without certain data, these omissions and other assumptions must be noted with the results of an analysis so there will be no misunderstanding in it interpretation.

To perform capacity planning functions, certain performance data is required. Six categories of data usage has been outlined in Figure 23. In column one labelled "System State", there are six basic parameters to be analyzed. It is felt that an analysis of these parameters will given an initial indication of the present state of the installation from a performance point of view. These are also the parameters to be forecasted. Initially, these points should be forecasted from data collected from some periodic tracking scheme. As such forecasts are validated, simple analytical queueing models may be integrated into the process as an additional predictive tool. The importance of tracking projections can not be over emphasized.

In an environment, where models are being developed and performance tracked, a "well tuned" system is essential. In column 2 of Figure 23 certain tuning parameters are indicated. Establishing a given level of tuning is something to be done on an as needed basis. Therefore, certain parameters analyzed during this process would not be collected in a tracking mode (as shown on Figure 23). In a batch environment, job turnaround time is the critical parameter relative to user satisfaction. Hence, certain job scheduling data must be tracked as shown in the last column of Figure 23.

The accuracy of the measurement tools discussed in this section is impacted by system hardware and software changes. Therefore, a user must think in terms of regular calibration schedules for tools. This means having standard job streams

MEASURED DATA / MEASUREMENT TOOL	C2	A2	CA6	CA7	CA8	CA12
LOADING						
• TRANSACTION TYPES			X			
• BYTES/TRANSACTION (IN & OUT)			X			
• TRANSACTIONS/SECOND/TERMINAL			X			
• LOGICAL FILE ACCESSES/TRANSACTION*			X			
• EXCP'S/DEVICE	X	X				
• TOTAL EXCP'S BY CHANNEL	X	X				
• BLOCK SIZE BY FILE						X
SERVICE						
• AVERAGE CPU TIME/TRANSACTION			X			
• ELAPSED TIME/TRANSACTION			X			
• TOTAL CPU TIME			X			
• TOTAL ELAPSED TIME			X			
RESOURCE UTILIZATION						
• SHORT-ON-STORAGE			X			
• MAXIMUM TASKS			X			
• STORAGE UTILIZATION				X	X	

* MUST IDENTIFY ACCESS METHOD USED

Note:
- C2 - SMF
- A2 - SGP
- CA6 - CICS Performance Analyzer II
- CA7 - CICS Plot
- CA8 - CICS Dynamic Map
- CA12 - Utility IEHLIST

Figure 21. CICS Capacity Planning.

MEASUREMENT TOOL	C2	A2	C5	A5	A6	CA9	CA10	CA12
MEASURED DATA								
LOADING								
• TRANSACTION TYPES			X		X			
• BYTES/TRANSACTION			X		X			
• TRANSACTIONS/SECOND/ TERMINAL			X		X			
• TRANSACTIONS/SECOND/LINE			X		X			
• LOGICAL FILE ACCESSES/ TRANSACTION						X	X	
• PHYSICAL FILE ACCESS/MPP (BATCH)	X	X				X		
• EXCP's/DEVICE	X	X						
• TOTAL EXCP'S BY CHANNEL	X	X						
• BLOCK SIZE BY FILE								X
• TRANSACTIONS BY MPP					X			
• MPP BY ADDRESS SPACE						X		
• NUMBER OF MFS PREFETCH I/O's						X		
• NUMBER OF MFS IMMEDIATE FETCH I/O'S						X		
• NUMBER OF MFS DIRECTORY I/O's						X		
• NUMBER OF I/O'S DUE TO INSUFFICIENT MESSAGE QUEUES						X		
• TOTAL NUMBER OF TRANSACTIONS					X		X	
• TOTAL NUMBER LOG RECORDS								
SERVICE								
• TOTAL CPU TIME/TRANSACTION			X	X				
• CPU TIME/TRANSACTION (MPP)						X		
• TOTAL ELAPSED TIME/TRANSACTION			X	X				
• ELAPSED TIME/TRANSACTION (MPP)						X	X	
• SCHEDULE TO 1st DL/1 CALL								
• CPU TIME						X		
• ELAPSED TIME						X		
• ELAPSED TIME (BATCH DL/1)							X	
RESOURCE UTILIZATION								
• MAXIMUM MESSAGE QUEUE SIZE						X		
• UNAVAILABLE BUFFER POOL SPACE						X		
• PROGRAM DEADLOCK OCCURANCES						X		

Note:
C2 - SMF
A2 - SGP
C5 - SIR
A5 - IMS/VS Log Transaction Analysis
A6 - IMS/VS Statistical Analysis
CA9 - IMS/VS Monitor Report Print Program
CA10 - IMS/TRAPDL1
CA12 - Utility IEHLIST

Figure 22. IMS Capacity Planning.

PERFORMANCE DATA	SYSTEM STATE	TUNING LEVEL	MODEL VALIDATION	TRACKING		
				TUNING	PROJECTIONS	SCHEDULING
LOADING (BATCH, ON-LINE)	X	X	X		X	
RESOURCE UTILIZATIONS	X		X	X	X	
RESPONSE TIMES	X		X		X	
TURNAROUND TIMES	X		X		X	
AVAILABLE FRAMES COUNT		X		X		
PAGING RATES (IN/OUT)		X		X		
SWAPPING RATES (IN/OUT)		X		X		
NUMBER OF INITIATORS	X					
AVERAGE NUMBER OF ACTIVE INITIATORS	X		X			
CPU, CHANNEL OVERLAP		X		X		
JOB START TIMES						X
JOB COMPLETION TIMES						X
RESOURCE UTILIZATION BY JOB						X
THROUGHPUT						X
JOB PRINT START TIMES						X
JOB PRINT COMPLETION TIMES						X
LINES PRINTED (BY JOB, TOTAL)						X
DASD SEEK ANALYSIS		X				
DASD CONTENTION ANALYSIS		X				
SVC ANALYSIS		X				
BUFFER ANALYSIS		X				
VS ADDRESS ANALYSIS		X				
STORAGE UTILIZATION ANALYSIS		X				

FIGURE 23. PERFORMANCE DATA USAGE (DATA).

available against which measurements are obtained. It is advantageous to use more than one tool to collect certain critical performance parameters for validation. Then, as the confidence in a particular tool is achieved this redundant collection can be terminated.

The performance measurement tools are a very important and integral part of the capacity planning effort. But its important must be placed in the proper perspective with respect to the other factors influencing capacity planning. One of the basic functions of capacity planning is to make equipment capacity predictions or forecasts (Figure 24). Obviously, as shown in Figure 24, measurement tools are key but the "Detail Analysis" phase is the most important aspect of this prediction/forecasting cycle. As shown, the measurement tools are used to collect current system performance data (Results 1) as well as data necessary for model development. The model (queueing, empirical, statistical) is then used to make predictions or forecasts (Results 2). As an indication of some initial capacity planning forecasting and predicting techniques, see Section 3.0 (Phase I Implementation). The model output is shown being compared to current measured data. It must be understood that in most instances Results-1 and Results-2 may be quite different. Then, only through careful and detailed analysis of the measurement tools, software subsystems (TSO, IMS, CICS, etc.), hardware subsystems (CPU, channels, control units, etc.) and the modelling technique will the inconsistencies in results be resolved. This means many different skills must be co-ordinated and brought to bear on this problem. Then, as shown in Figure 24, resolution of problems will result in modifications to the current model and new predictions/forecasts are made. After each iteration, projections should improve. The importance of the detail analysis phase cannot be over emphasized at this time.

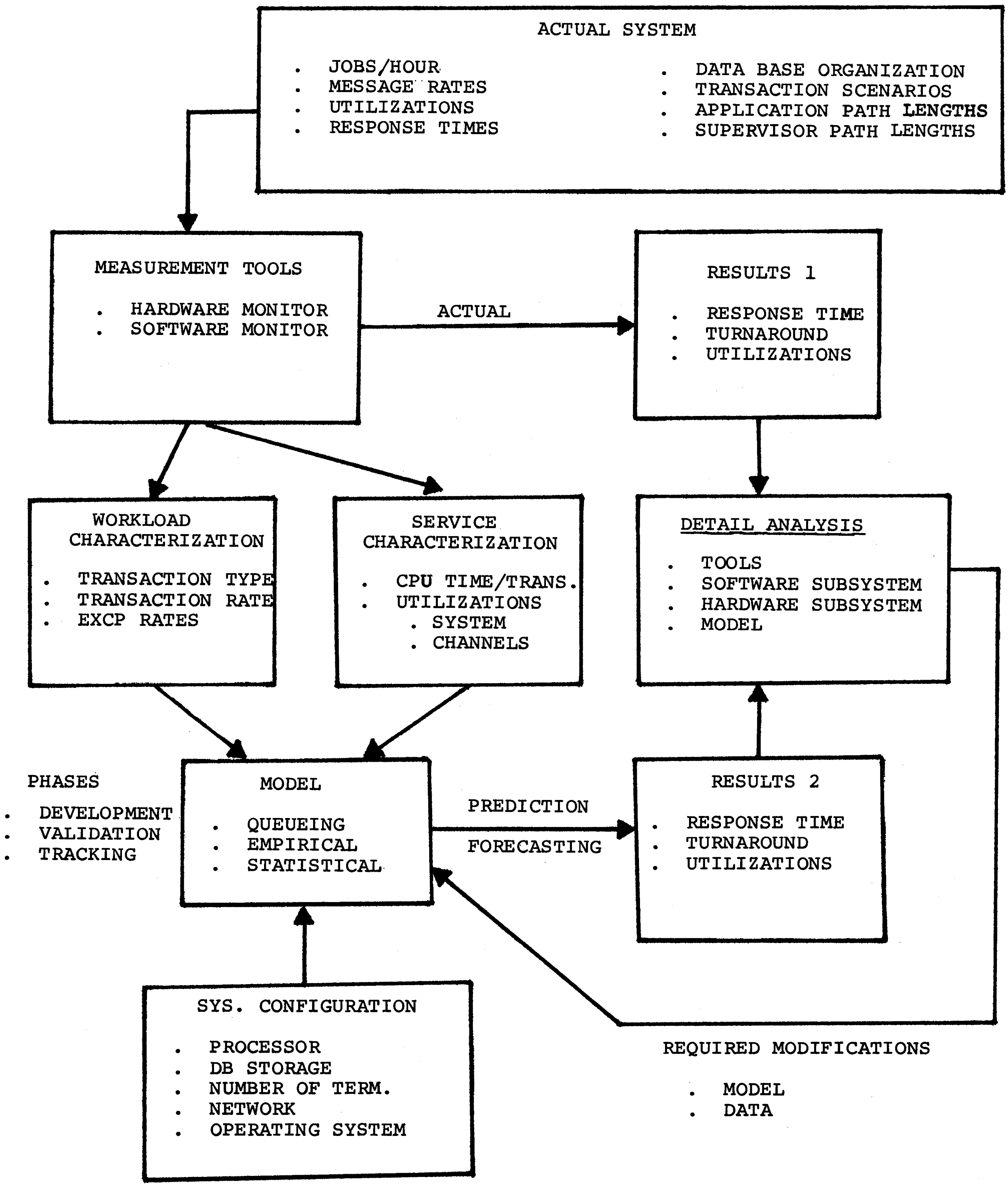


FIGURE 24. PERFORMANCE PREDICTION/FORECASTING CYCLE.

3.0 IMPLEMENTATION

3.1 INTRODUCTION

Implementation of a capacity planning program in a data processing installation probably will not follow the same format for all users. Since capacity planning is a term for a process which has been performed in varying degrees of depth by many users over the years, some installations will be able to implement more detailed portions of the methodology than those just initiating a program. For example, it makes no sense to begin to implement very detailed modelling schemes for prediction (i.e., GPSS, CSS) when data gathering techniques are crude and ineffective. Confidence in model results can be no greater than that of the input data. Therefore, the assumptions for the implementation process discussed in this section are that the installation has had no prior capacity planning experience. This means that those users with varying levels of experience may pick-up the process at the point most applicable to their installation.

As discussed in the following sections, capacity planning should be implemented in four phases as outlined below.

PHASE	SUBJECT
I	ESTABLISH CURRENT SYSTEM PERFORMANCE LEVELS
II	ESTABLISH USER SERVICE OBJECTIVES
III	INTEGRATE SCHEDULING SCHEME
IV	SYSTEMIZATION

This particular phased approached does not necessarily have to be rigidly adhered to, but the reasoning is that a total capacity planning effort is too large to implement without definite milestones and benefits reasonably spaced throughout the overall process.

3.2 ESTABLISH CURRENT PERFORMANCE LEVELS (PHASE-I)

It is very important in the initiation of a capacity planning effort not to disrupt any existing computer performance evaluation (CPE) programs. The measuring

techniques and the parameters being measured should be clearly understood. It is necessary to understand the current program because the personnel involved as well as much of the data being gathered may naturally interface with a capacity planning effort. Also, any experienced performance personnel should become an integral part of the capacity planning program.

In the initiation of this effort, data requirements should be kept to a minimum. Hence, the primary parameters to be measured are:

- Workload
- Resource utilization
- User service
- Available capacity
- Unattainable capacity.

Although Phase-I is primarily devoted to performance, this phase should also be devoted to developing a clearer understanding of the operation and scheduling of jobs/workunits through the installation. A prime factor in this analysis is a clear understanding of the total workload. In acquiring this understanding, the workload should be broken down by application areas. For example, the customer order processing application area may be basically a batch operation and include several jobs. Then, it should be established from the total number of installation batch jobs (workunits) and CPU job-hours (service), how much is attributable to the customer order processing application. This break down of the workload should be carried through for each of the major application areas. Hopefully, most of the total load will be accounted for in this process. There will probably be a need to establish a miscellaneous category for those jobs processed which do not belong to a specific application area.

The utilization of the various resources (CPU, channels, I/O devices) must be established. These values are collected on a continuing basis to establish any trends or possibly any correlation with system workload. The process of collecting data over some period of time tends to improve confidence in measurement techniques as well as in the data collected. In monitoring utilization, it is very important that peak periods as well as heavy users be identified.

In many instances, the peak periods are the critical times of the day and system design must be such that user service objectives are adequately met. Heavy users are monitored because of the large amount of resources consumed. Also, these jobs are prime candidates for program tuning. It has been found, in many studies, that tuning the jobs which consume a large portion of each resource has greater return than normal system tuning (operating system, channel load leveling, DASD data set placement, etc.).

During this phase, those factors (system down time, operational overhead, etc.) that contribute most to unattainable capacity should be identified. Capacity planning can only be effectively accomplished when it is known how much capacity is available for scheduling and future planning (predictions/forecasting). As the capacity planning effort continues and other factors noted that act to reduce overall capacity, available capacity figures may be altered.

As stated earlier, tracking performance is an integral part of capacity planning. Tracking implies continuous system measurements, hence, attention to the level of system loading (overhead) imposed by a particular measurement tool is very critical. This means, it is practical to have certain tools (low overhead) continuously monitoring system performance. When it becomes absolutely necessary to use high overhead type tools, it is best to restrict their use to samples at small intervals during production time. For example, the GTF (Generalized Trace Facility) measurement tool, in most instances will impose high system overhead. But, for certain severe problems, the level of detail required can only be captured by GTF. When this is the case, the tool should only be applied for small intervals of time (5-10 minutes) during the production shifts.

Also, Phase-I is the time to evaluate the type and number of reports required to support the capacity planning program. The number must definitely be kept to a minimum. Simplicity and readability is paramount. This implies that graphical reports will definitely be used. Report formats and content will vary between installations.

An example illustrating the initiation of a capacity planning program is outlined on Figure 25. The installation contains a 370/168 with the MVS operating system. The primary subsystems are batch, TSO and CICS. Only three

- Example system
  - MVS (Batch, TSO, CICS)
- Measurement Tools
  - MF/1-RMF
  - SMF
  - CICS performance analyzer
- Identify Major application groups
  - Sales, 9 Jobs (1 major CICS application)
  - Accounting, 10 Jobs
  - Data Processing
  - Testing (some TSO)
  - Operations
  - Systems programming (some TSO)
  - Operating System
  - Miscellaneous

  Note: Application CPU resource consumption listed in job-hours/time period.
- Data to be collected by shift (physical/logical)
- Loading (meaningful, only if accompanied by additional service data, long processing, short process, etc.)
  - Batch (job rate by application grouping)
  - TSO (ended transactions, TGETS/time period)
  - CICS (transaction rate by type)
  - Channels (EXCP rate)
  - Device (EXCP rate)
- Utilization (CPU Time)
  - Total (elapsed-wait)
  - By application group (job hours)
    - Batch
    - TSO
    - CICS

Figure 25. Capacity Planning Example.

- Utilization
    - Channels
    - I/O Devices

- Batch turnaround times
- CICS response time
- TSO response time
- Determine peak periods
- Identify data holes

- Data Analysis
    - Validation
    - Identify and analyze trends
    - Correlation
    - Load
    - Utilization
    - Response/turnaround time

- Begin to make gross projections from trend data and validate projections (Figure 28)
    - Load increases
    - Required service (CPU job-hours)
    - Resource Utilizations

- Identify new projects planned for implementation during next 24 month period

- Make gross projections on new applications
    - Load increase
    - Required service (CPU job-hours)
    - Resource utilizations

- Begin to establish guidelines for projections.

- After a certain level of confidence is established in measured parameters
    - Loading
    - CPU service (application job hours)
    - Resource utilization
    - Response/turnaround times

One may begin to move to simple queueing models which will hopefully enhance projections.

Figure 25. Capacity Planning Example (Cont'd.).

measurement tools are recommended. Because of the current state of the art of measurement tools certain limitations must be accepted. Major measurement tool deficiencies must be identified and supplementary actions outlined. For example, CICS performance analyzer calculates only that portion of a transactions response time spent in the CICS system. The remainder of the response time must be approximated and validated (e.g., by a stop watch). One of the primary purposes of this initial effort is to establish some confidence in certain basic data (See Figure 25) which these tools collect.

A first step in capacity planning is the understanding of the system workload by application area (sales, accounting, data processing, etc.). This categorization might be made by department (engineering, research, etc.) or whatever category is most appropriate for your installation. One of the primary reasons for this categorization is analysis and projection. Each application area consumes a certain amount of the data processing resource. The CPU, one of the primary resources, is consumed by what is termed "job-hours". A job-hour is an hour of CPU processing or busy time (Figure 4) required to service the workload (programs, transactions, etc.) being input from various application areas. The number of job-hours consumed by each application area during some time interval (day, week, month) should sum to the total number of job-hours of CPU busy time during that period.

Measurement tools are not available to automatically segment total processing time into specific application times, therefore, care must be taken in the division of total processing time (Figure 26). The current tools proposed to segment processing time are:

- MF/1 - RMF
- SMF
- CICS Performance Analyzer.

Basically, MF/1 or RMF will report on overall system activity. The primary outline of Figure 26 will be developed from CPU wait records reported by RMF. Then, total period or elapsed time minus wait time gives CPU busy time for calculation of CPU utilization. The primary purpose of the other tools listed above is to segment this total utilization by application area. The tool most used

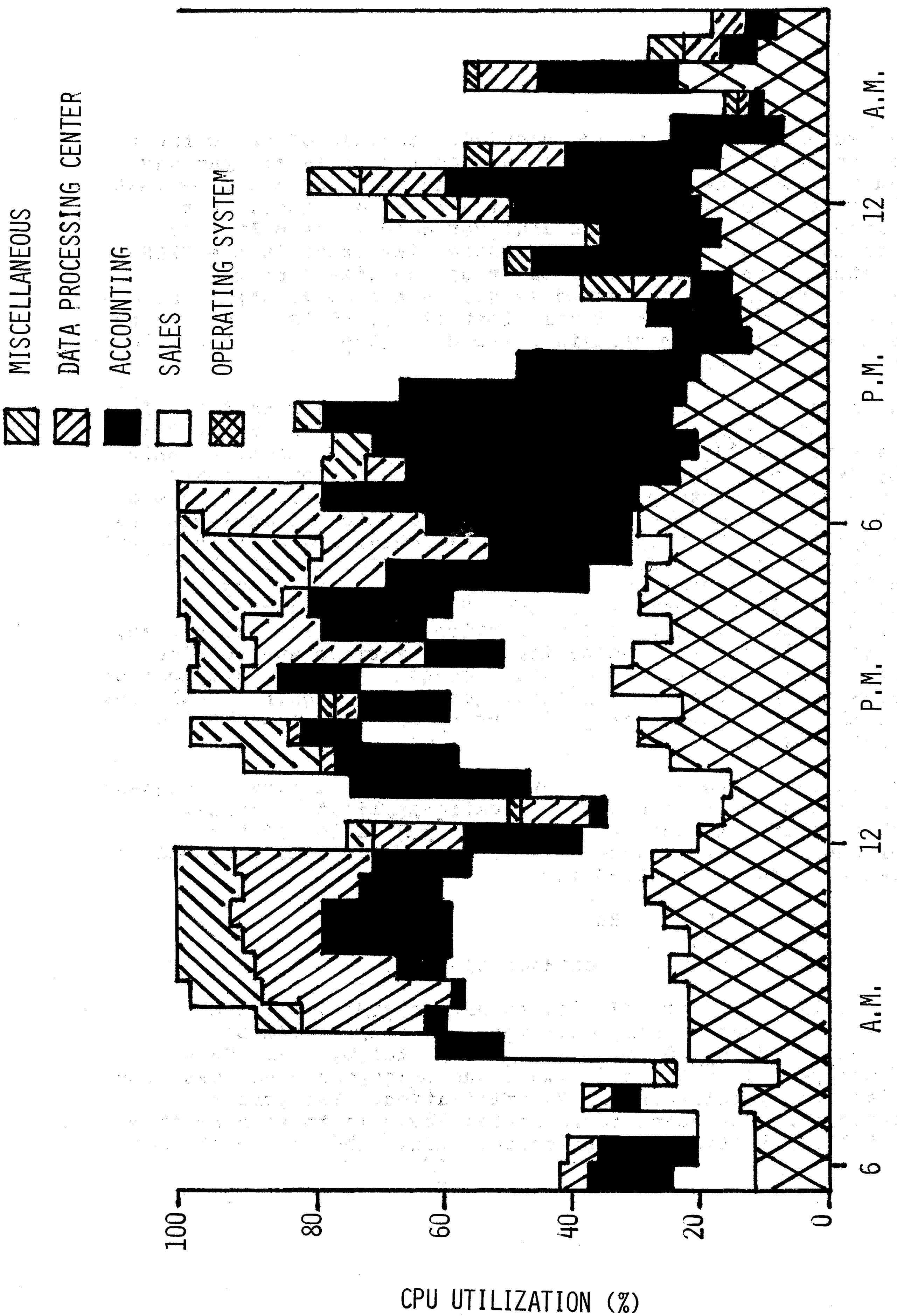


FIGURE 26. CPU CONSUMPTION BY APPLICATION.

for this function is SMF. SMF collects the CPU job-hours consumed for all application jobs started via a job control card. SMF will not collect data on jobs started from the console. SMF collects data for a given job mix while the operating system is running under a job's TCB. The ratio of job TCB to total CPU utilization in not a constant. For example, Figure 27, which is the result of measurements made at a users installation, shows the variability of SMF job-hours in relation to total CPU job-hours consumed at a specific CPU utilization. The differences in the ratio of SMF job-hours to total CPU job-hours are a function of many things. For example, certain SVC service time is not recorded by SMF under the SVS operating system. The characteristics of the jobs being recorded affect the resultant job-hour value. The characteristics considered are concerned with job running time (short vs long) and the amount of I/O activity which is related to the amount of supervisor services required. These facts concerning the use of SMF data are discussed to establish a need for a factor to convert from job-hours to total CPU hours. In such an analysis, it should be understood that SMF reporting is making it possible to organize and better understand our data processing environment from an applications point of view. We are well aware of the accuracy of SMF data. But it is felt that an installation will profit from this organization and understanding process. Keep in mind, this is an initial step in the planning process and more detailed analysis techniques are available for users already at this level.

For the installation referenced above, a factor for modification of SMF data may be developed, Figure 26 indicates that a ratio of SMF job-hours to total job-hours between 0.4 and 0.55, where utilization values range between 50 and 75 percent, is grossly representative of the account workload. Hence, as a first approximation of the actual job-hours ("problem program" and "supervisor") consumed by each application group, the SMF job-hours reported must be multiplied by a factor somewhere between 1.82 and 2.5. This factor will vary between accounts, but the major point is to arrive at a factor with which you are satisfied. The net result is to have application SMF job-hours times a ratio which will sum to the total system job-hours. Then, using this as a base and making modifications where appropriate, SMF job-hours can be tracked on a continuing basis.

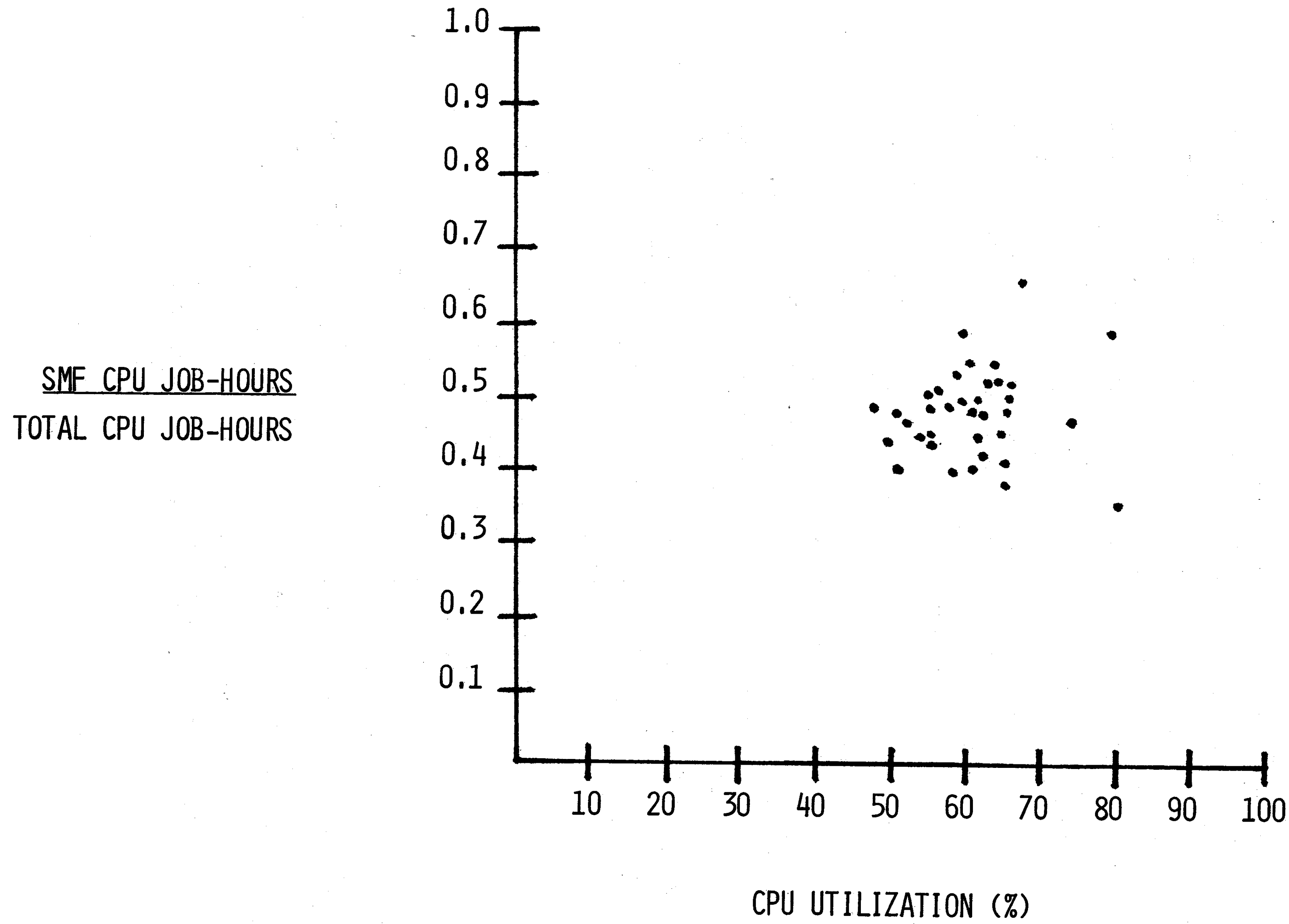


FIGURE 27. SMF VERSUS TOTAL CPU JOB-HOURS.

As indicated on Figure 25, other data will be collected (loading and service) to adequately evaluate the data processing environment. The job-hour parameter addresses the CPU, but channel and I/O device loading and service are also a critical part of the analysis process. The primary point of capacity planning is the management of all data processing resources. The data collected must be analyzed and validated. For example, can the installation make reasonable workload growth projections for the accounting application area from past history and projected new applications. By gathering current data, analyzing trends, making gross projection from these trends and validating, an installation will begin to attach a certain confidence to their projections (Figure 28).

In order to make projections for new applications and improve their predictive capabilities, installations must identify these at least 12 to 24 months in advance. Then, they can make gross projections based on their current workload assessments. For example, certain programs in the accounting area are consuming a certain number of SMF job-hours, attempt to quantify new application in the same way. Using the job characteristics in a gross way to make these predictions. The primary return from this activity comes when the application goes on-line and the projection is analyzed. This will hopefully expose certain dificiencies in the technique. Correction of these deficiencies should make future projections better. Also, out of such a feedback process basic guidelines will be established.

To some readers, this approach to capacity planning might seem too basic. Let me assure you that such a basic process is very necessary in many user situations. Those DP managers who understand their current workload and service requirements from an applications point of view, you are tracking performance on a continuing basis and have identified certain performance trends and can talk with confidence about future workloads and system service requirements. This confidence in data has probably been established with SMF data as well as other supplementary data gathered with more sophisticated tools. An account at this level is a prime candidate to begin to move toward single server queueing analysis (Section 4.0) as a means of enhancing your predictive techniques. This technique is still grcss requiring approximations, guidelines, rules of thumb but it begins to move the analysis into the analytical

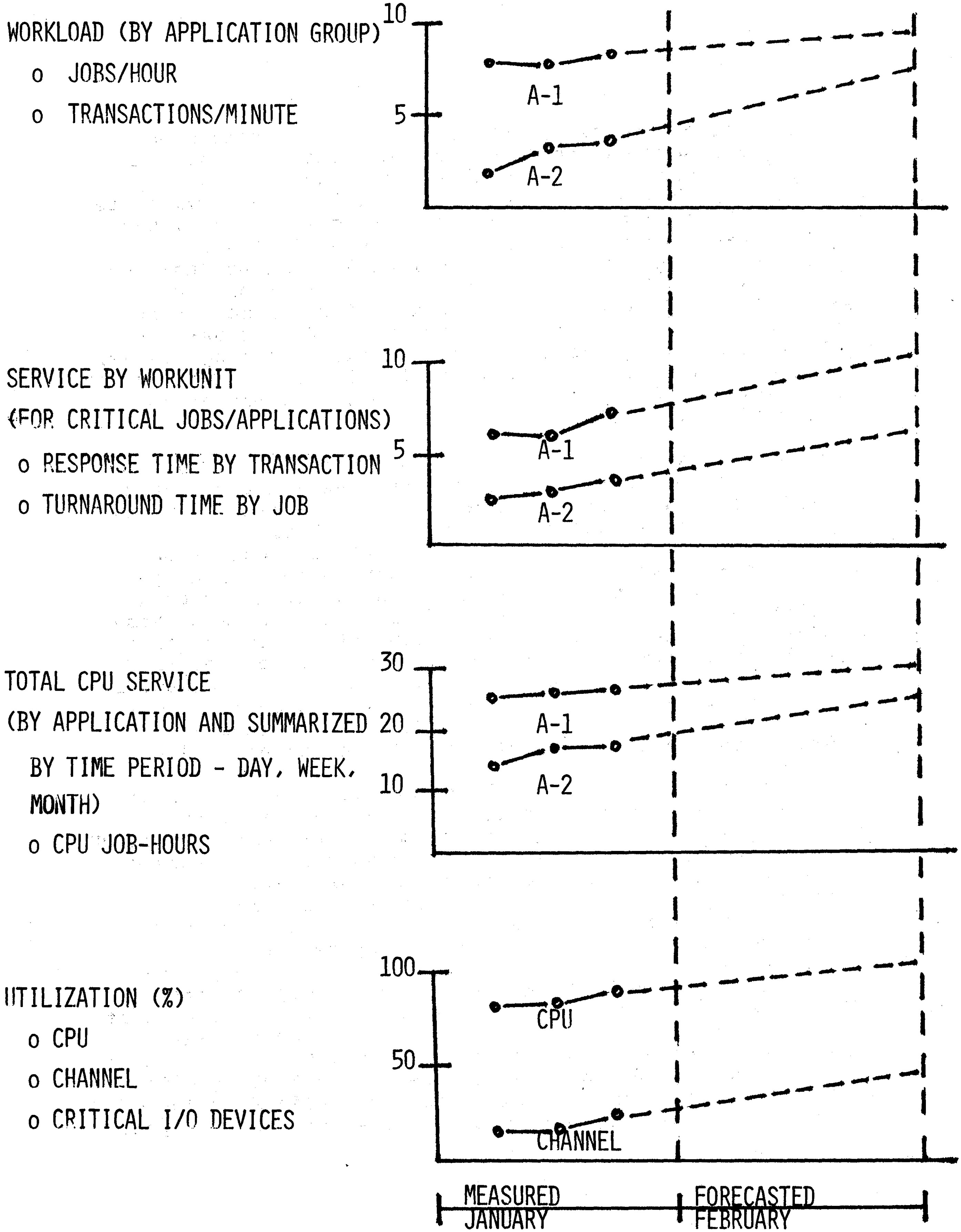


FIGURE 28. PERFORMANCE FORECASTING

realm. More sophicated techniques (e.g., closed model queueing, numerical models, simulation, etc.) are available as certain single server analytics become insufficient. The major problems in complex computer system analysis, characteristically, has not been the available analytic techniques, but it has been an understanding of the system relationships (hardware and software).

3.3 Establish user service objectives (Phase-II)

The user service requirements, as discussed in Section 1.6, are the key to system capacity. Therefore, it is very important as an early part of the capacity planning effort that user service objectives are quantitatively established and agreed upon. The agreement is obtained between operations and the various users. It is the critical service requirements that tend to define the upper limits of system capacity (See Section 1.6).

It is necessary to track service parameters to understand the relationships that exist between workload resource utilizations and service (response/turnaround time), as they relate to overall system capacity. Service must be tracked to indicate the level of service being given so that corrective action can be taken before users become dissatisfied. Tracking of service objectives is a continuing function.

3.4 INTEGRATION OF SCHEDULING SCHEME (PHASE-III)

Scheduling is a very important part of capacity planning. Hence, a major effort must be applied to understanding the current scheduling scheme. In Phase-I, resource performance levels (loading and utilization) were monitored and recorded. Part of the Phase-III effort is to determine what users are responsible for the system workload and the time of day their load is applied. In essence, this is an integration of performance and scheduling profiles. Also, any system bottlenecks should be identified during this phase and proper corrective actions taken. For example, loading trade-offs when channels are posing a bottleneck, addition of new fixed head DASD sevices or certain software updates.

3.5 SYSTEMIZATION (PHASE-IV)

The final phase of implementation is basically an overall review of the capacity planning effort. At this point, performance tracking is evaluated to establish that all required parameters are being measured. The current reporting process is evaluated to make sure only the necessary reports are being generated and the formats are acceptable. In the reporting area, a historical file should be set up for recording pertinent historical performance information. This file will serve as the basis for developing performance guidelines and rules of thumb. For example, a new megabyte of main memory is purchased and installed. This installation would probably affect paging values as well as user response times. These factors should be recorded and used to aid in evaluating a main memory problem at some later point in time.

4.0 Performance Prediction

4.1 Introduction

The approach to performance prediction developed in this bulletin is primarily empirical. Through measurement and analysis of software and hardware properties, system insights will be gained and empirical relationships developed. Also, guidelines and rules of thumb for system analysis will be established. Very important to the process of analyzing computing systems is a basic understanding of certain software and hardware saturating properties. Saturation is indicative of severe bottlenecking. In some cases proper analysis and adjustments may relieve or even remove a bottleneck, whereas other bottlenecks are a permanent part of the software or hardware and can be removed only by redesign. A very good example of software saturation of the permanent type is the 200 byte DOS transient area. For application programs with large I/O loads, the transient area become a bottleneck and restricts overall CPU utilization. Projections using DOS in some environments meant the CPU could only be driven to 50 per cent utilization. For capacity planning purposes, it is essential to know that a resource as critical as the CPU can only be used to 50 per cent capacity. Hardware saturation properties are equally important, for example, channel utilization of 35 per cent causes excessive response time for RPS DASD devices. Capacity predictions would be grossly inaccurate if it was assumed that channels could be driven at 100 per cent utilization.

In general, performance prediction may be viewed as shown in Figure 29. Having some projected load (new or increased current workload) and some user prescribed performance threshold, a system performance curve (Conf-A) is projected. This projection may be made from data trends, results of single server queueing models, or more sophisticated means if the situation dictates. The primary input to these models is the loading curves. From a loading standpoint, the curve might show transactions/second, jobs/hour, etc.

From a loading point of view, many installations may not make loading projections in transactions/second or jobs/hour. They forecast their load increase in check volumes for banking, increase in automobile or tractor production for manufacturing or principally in the goods or

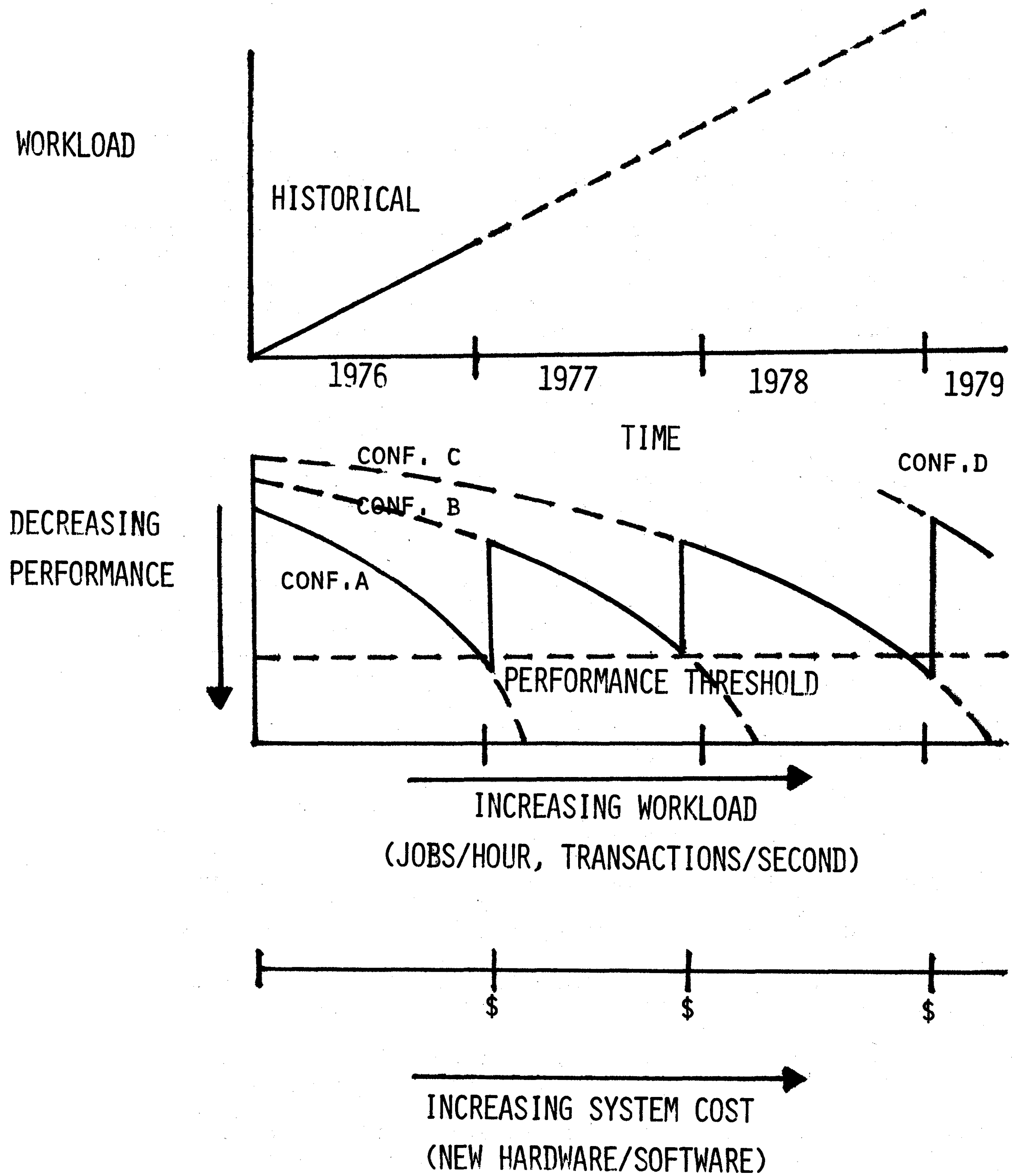


FIGURE 29. PREDICTION/FORECASTING PROCESS.

services produced. The problem then becomes one of translating these volumes to computer load volumes. It appears that making this kind of translation will be very difficult for those installation not tracking and correlating product loads with current data processing loads. Then, as product forecasts are made, data processing projections can be made in the same current relationships.

Returning to the general performance prediction problem above, the performance threshold might be a 3 second response time not to be exceeded in a on-line environment or a 10 minute batch turnaround time. As shown in Figure 29, the capacity planning process would be one of selecting a load value from the curve (shown as being linear only for example purposes) calculating certain performance parameters indicative of some point on the configuration (Conf-A) performance curve. As the load increases, system performance moves along the Conf-A curve until performance degrades below the threshold. At this point, recommendations are made for configuration changes which move performance back into the satisfactory range. These recommendations might indicate a change in CPU, upgrading a movable head storage device to fixed or addition of another block multiplexor channel. This new configuration is indicated as Conf-B in Figure 29. This process would continue on out in time (1976-1979) to the last configuration (shown as Conf-D). In section 4.4, this prediction process is addressed in greater detail.

4.2 Motivation for a simple method of performance analysis/prediction

There is a very strong demand to find a simple, fast method of doing performance analysis/prediction. Using single server queueing models, along with certain basic guide lines and rules of thumb, it is possible to perform relatively fast, simple system analysis. These guidelines, which will vary for each analysis, can be developed from close observation and measurement of various system parameters (workload, resource utilization, response time, etc.). It should also be understood that the development of a single server approach to the analysis of complex computer systems composed of very complex queueing relationships is not theoretically justified. However, from a practical point of view where these simple models are supported by empirical analysis and systems experience, the approach is

well justified. In practical situations, this method is motivated by the following heuristic arguments:

1. In many cases, the source data on workload is extremely scarce. The workload profile, for the most part, is an educated guess. It does not pay to use any complex model in order to achieve great "accuracy" in the results.

2. The characterization of system components (CPU, channels, I/O devices, etc.) involves many simplifications. Therefore, no matter how detailed the model for the system, the results of analysis would be of a gross nature.

3. The requirement for the results of a performance analysis may not allow time for a detailed analysis which would be necessary for numerical models or simulation.

4. The tools and facilities to perform a detailed analysis are not always available.

5. In many instances, grossly approximate results are quite acceptable if more iterations and refinement of the analysis are to follow.

6. In certain cases, it may be found that more detailled analysis takes a great amount more of time and effort but may yield only a slight improvement in accuracy of the results.

4.3 Queueing Analysis

The next two sections are intended as a very brief introduction to queueing analysis. For a more detailed discussion of queueing analysis see Reference 1. The primary purpose of this section is to introduce and discuss certain basic queueing terminology. Most analytical computer models for system performance analysis are built upon a branch of applied probability theory known as "Queueing Theory". Queueing Theory is also known as Traffic Theory, Congestion Theory, Theory of Scheduling, or Theory of Stochastic Service Systems.

In the performance analysis process, the computer system becomes a queueing system or network of queues. A queueing system consists of a source of potential customers, one or more waiting lines, and one or more servers. A customer is one who uses the service facilities. In computer systems, a customer may be a job, a transaction, an application program, an inquiry, an I/O request, etc. A server is a facility which provides service to the customers. The server may be a CPU, a channel, a transmission line, or an I/O device.

A queueing system may be described by the Kendall [2] notation which has the form

A/B/c

Where A = Interarrival time distribution
B = Service time distribution
c = Number of servers.

The various symbols that may be used for A and B are the following:

G or GI = General Independent Distribution
M = Exponential or Poisson Distribution
Ek = Erlangian-K Distribution
D = Deterministic (constant) Distribution.

For example, a

M/M/1

system has a Poisson interarrival time distribution, exponential service time distribution and a single server. Also, some very important system characteristics implied by this notation is that there is an infinite source population of customers, nc limit in waiting line size and service is given on a first-come first-served (FCFS) basis.

When necessary, additional notations may be appended to the above description as

A/B/c/K/n/Z

where K = The limit on the number of customers possible in the system

n = The limit on the number of customer possible in the source

Z = Queue discipline.

The following are the most often used queueing disciplines:

- FCFS, First Come First-Served
- LCFS, Last Come First-Served
- RSS, Random Selection For Service
- PR, Priority.

For example, a

M/E3/2/10/100/FCFS

system has a Poisson interarrival time distribution, Erlangian-3 service time distribution, 2 servers, a system customer limit of 10 (that is, 2 in service and a maximum of 8 in the waiting line), a source population of 100 and a first-come first-served queueing discipline.

In a priority system, customers are divided into priority classes. Customers in a high priority class have preference over customers in all the lower priority classes. Customers in the same priority class are usually served in order of arrival, FCFS. When a customer of a high priority class arrives at the system and finds another customer of a lower priority class in service, there are several possible control policies. In a non-preemptive priority system, the newly arrived customer waits until the customer in service completes, then he is allowed access to the server. In a preemptive priority system, theoretically service is interrupted immediately (as soon as possible from a practicle system point of view), then the newly arrived high

priority customer begins service. After completion of his service, if there are no customers of higher priority in the system, the low priority customer whose service was interrupted continues his service. In a preemptive resume priority system, the low priority customer resumes his service at the point of interruption upon his next access to the server. In a preemptive repeat priority system, the lower priority customer repeats his service from the beginning at the next access to the server.

4.4 Single Server Queueing Models

The single server queueing model (Figure 30), which is the simplest of the queueing systems, is the basis for making gross estimates in complex computer system performance analysis. As shown in Figure 30, the two models used for analysis are:

- M/M/1
- M/G/1

The terms used in defining these models are outlined below. All quantities are given as average values.

L - Workload (transactions/sec.)

Q - Number of customers in queue (transactions)

U - Utilization (busy time/elapsed time)

Tw - Time spent waiting in the queue (seconds)

Ts - Time spent in service (seconds)

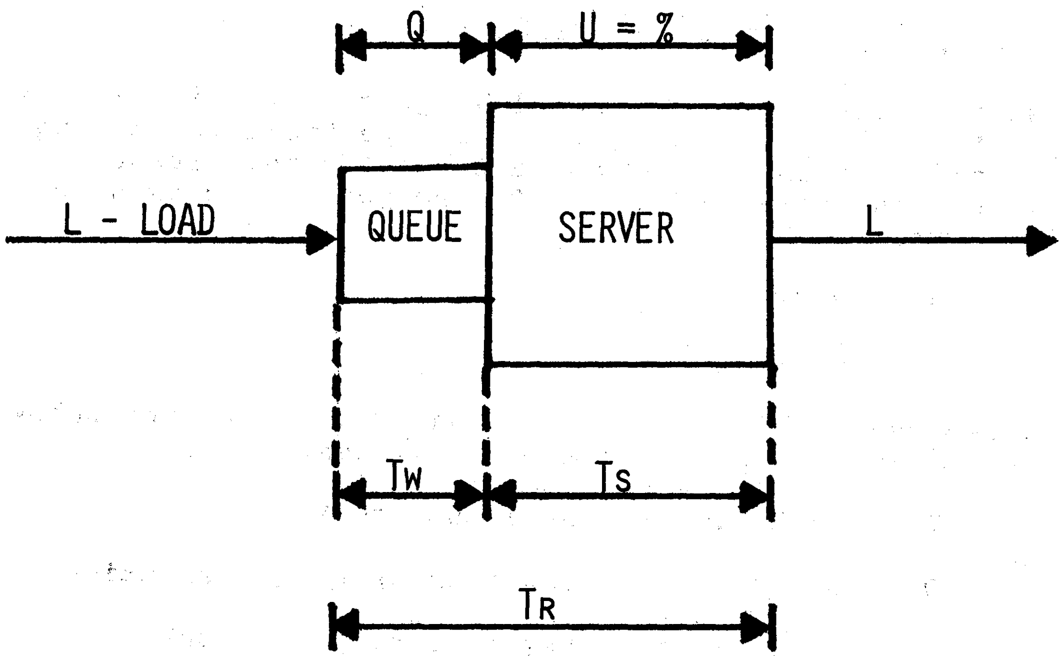
Tr - Response time (elapsed time, Tw + Ts)

VAR - Variance of service time (square of standard deviation).

In the analysis of a resource as a single server queue, the primary data required are loading and service. Then, it is possible to calculate waiting time (Figure 30), where the measured elapsed and response time values may be used for model validation. For example,

RESOURCES: CPU, CHANNEL, I/O DEVICE, ETC.

$U = LT_S$



M/M/1: $T_W = \frac{T_S U}{1-U}$

M/G/1: $T_W = \frac{U T_S}{2(2-U)} \left[1 + \frac{VAR_S}{T_S^2} \right]$

$T_R = T_W + T_S$

FIGURE 30. SINGLE SERVER QUEUEING MODEL.

L, Ts, Tr are measured

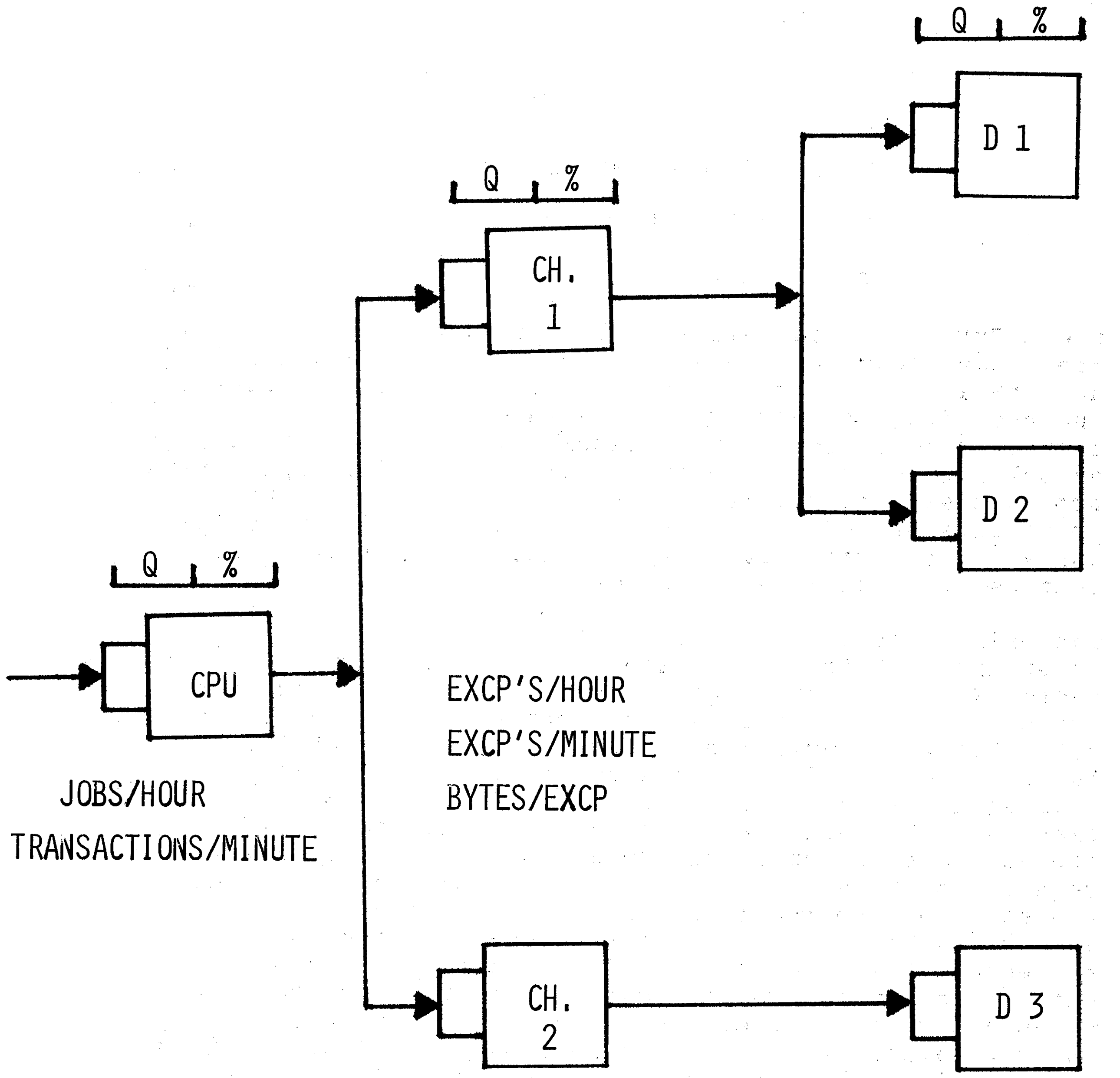
Tw is calculated

Tr* = Tw + Ts

where Tr* is the calculated response time to be compared to the actual measured response time (Tr). Be forewarned, Tr and Tr* in many instances are not equal. It will take an understanding of your system, measurement tools and technique to satisfactorily resolve the discrepancies. For example, using SMF to measure service time for application programs will result in inaccuracies because all the components of CPU service time are not recorded. In this instance, the wait time (Tw) and more specifically the response time will be incorrect.

The primary purpose of this analysis technique is to determine the time a transaction or batch job spends in the system (combination of many resources). Therefore, the time spent at each resource must be accumulated to determine this total time. A network of queues (Figure 31) outlines best the procedure applied to the total system. As shown, loading and service data are required at each server. For example, a CPU workload (transactions/min) generates a corresponding channel and I/O device load (EXCP's/min). The required service at each server is calculated by multiplying the service time required for each workunit by the workload and obtaining the total required service which is represented as a percentage of the total available time. With these parameters, it is possible to calculate response times and validate them. Reasonable accuracy for the current environment should be obtained before predictions can be made for future loads. Although this analysis technique has certain theoretical inaccuracies (i.e., staging of M/G/1 queues), the practical aspects of simplicity and necessity for indepth systems understanding (software and hardware) has made it a very viable approach for today's complex systems analysis.

In the analysis of the CPU, there are two other factors to be considered. These are priority scheduling and multiprogramming level. Normally each subsystem (Batch, TSO, IMS, etc.) is analyzed as though it occupies the highest priority level in the system. This means the queueing equations (M/M/1 or M/G/1) are used directly. For



NOTE: CONTROL UNIT CHARACTERISTICS ARE ALSO STUDIED AS PART OF THIS CONFIGURATION.

FIGURE 31. NETWORK OF QUEUES.

example, the response time equation for the highest priority level on Figure 32 is the equation used for the M/M/1 queueing system. As shown on this chart, the response time at lower priority levels is a function of the loading, service and utilization of the given level and all higher priority levels. For a more detail discussion of the formulations of Figure 32 see reference 3. Using these preemptive priority queueing equations, it is possible to analyze in a gross fashion a system as shown in Figure 33. The total system which was initially analyzed on a subsystem basis is further analyzed to account for the additional wait time imposed by higher priority subsystems.

The multiprogramming level of the CPU can be analyzed with the aid of Little's Law [4]. The law establishes a relationship between the average number of workunits in the system (N), the average number of workunits arriving per unit of time (L), and the average amount of time each work unit spends in the system (Tr). His law states,

$$N = L\ Tr.$$

In a batch environment, the parameters of this relationship may be defined as follows

- N - Average number of active initiators
- L - Average number number of batch jobs processed per hour (assuming steady-state workload, input is equal to output)
- Tr - Average elapsed time for a batch job in hours.

Bear in mind that analysis using Little's Law is approximate and in most instances will require analysis of other system factors; namely CPU utilization, job mix or contention problems, paging and main memory size.

4.5 Benchmarking

The most accurate and probably most costly method of analyzing and predicting the performance of a given system is to run the actual system under normal production load and assess its performance. For future planning purposes, current loads can be increased to the projected levels and the system performance evaluated. Although this approach is

$$T_r = T_w + T_s \quad , \quad U = L \; T_s$$

PRIORITY LEVEL - 1 (HIGHEST LEVEL)

$$T_{r1} = \frac{1}{1}\left[T_{s1} + \frac{T_{s1} U_1}{1-U_1}\right] = \frac{T_{s1}}{1-U_1}$$

PRIORITY LEVEL - 2

$$T_{r2} = \frac{1}{1-U_1}\left[T_{s2} + \frac{T_{s1} U_1 + T_{s2} U_2}{1-(U_1 + U_2)}\right]$$

PRIORITY LEVEL - 3

$$T_{r3} = \frac{1}{1-(U_1 + U_2)}\left[T_{s3} + \frac{T_{s1} U_1 + T_{s2} U_2 + T_{s3} U_3}{1-(U_1 + U_2 + U_3)}\right]$$

NOTE: LOADING, SERVICE AND UTILIZATION OF HIGHER LEVEL PRIORITY GROUPS WILL AFFECT LOWER LEVELS.

FIGURE 32. RESPONSE TIME FOR PREEMPTIVE PRIORITY QUEUES (M/M/1).

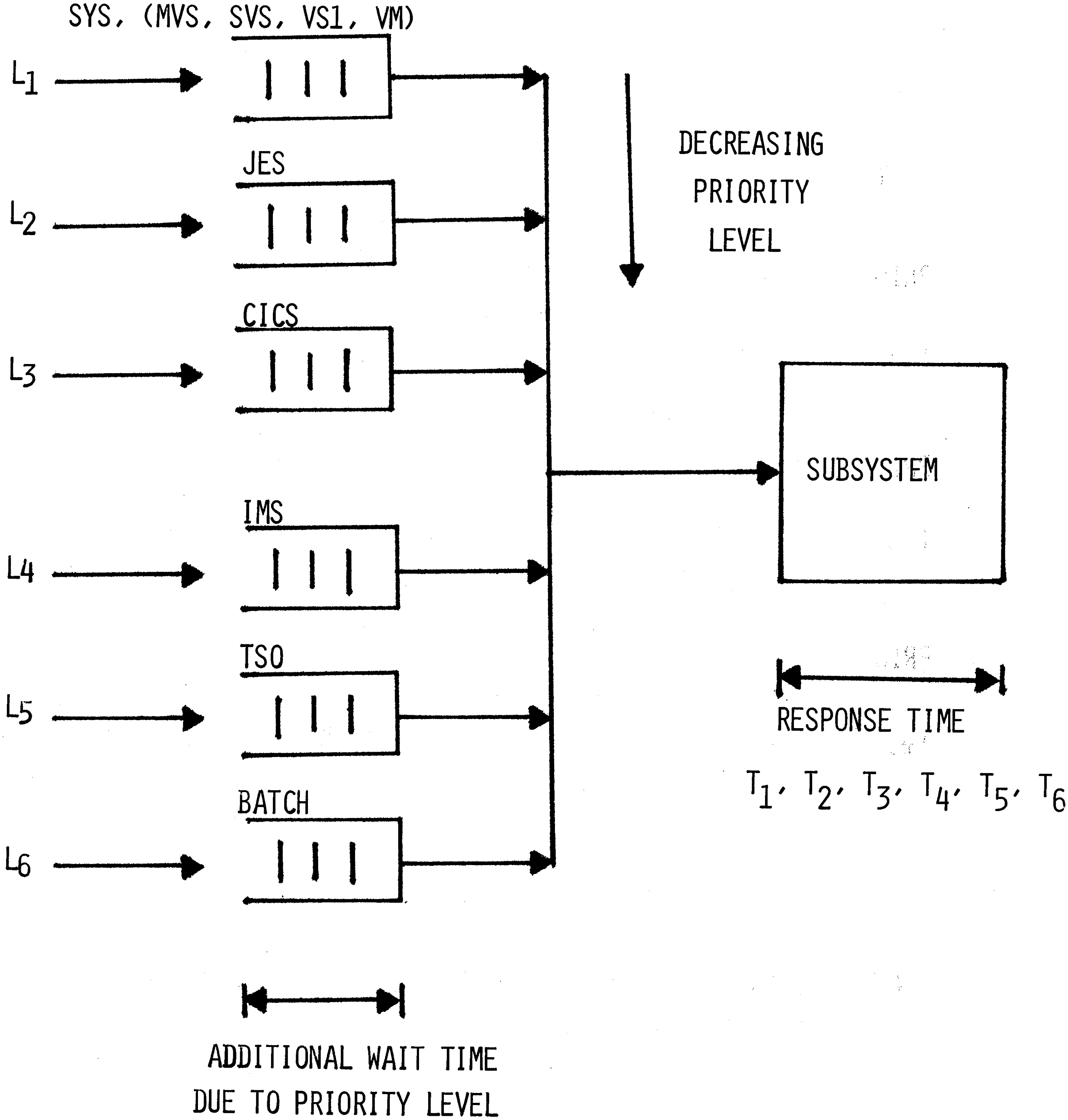


FIGURE 33. TOTAL SYSTEM ANALYSIS.

not always used, it proves useful in certain cases. The closer one gets to actual production operation the more accurate the results.

A benchmark is a compromise to total production operation and is the process of selecting and executing a portion of the production workload which "best" represents the real environment. Benchmarking may be accomplished on current software or hardware or on some proposed configuration. Performance assessment and prediction is the main purpose of this effort. For example, in Figure 34 a TP system is configured which has excess capacity. The question to be answered is, "How much growth in load is possible before this excess capacity is exceeded?" A system, as shown in Figure 35, might be configured to answer this question. Here, the modems have been replaced by a data set eliminator and the terminals by a CPU running the TPNS driver. Drivers are simply a set of programs used to simulate or replace a user defined terminal workload. It is necessary for the user to provide a terminal script of his environment. This simulation of terminal activity for performance analysis is transparent to the user application programs. Actual lines, communication controllers, etc. are all normally configured. The only restriction is that the system where the IBM driver resides must be attach to the 3705 controller (Figure 36). It is possible with a driver and its designated script to simulate a system in a controlled environment and gather the required performance data. Future environments or stress conditions (i.e., increased loadings, transactions/second) may be applied to the current configuration to establish workload limits. IBM has two drivers currently available these are the Teleprocessing Network Simulator (TPNS) Program and the Data Base Data Communication (DBDC) Driver.

The drivers may be used in either simplex or duplex mode (Figure 36). In simplex mode, both driver and application programs reside on the same CPU. This is primarily used for functionally testing application programs but limited performance analysis may be accomplished. It is possible to analyze changes in performance. Although the driver is exerting a certain load on the system, by analyzing performance at various load points certain driver loading effects may be removed. Duplex mode is recommended when detail performance analysis is required. In this environment, the driver occupies a separate CPU. It should be noted that stress testing requires the driver reside on a

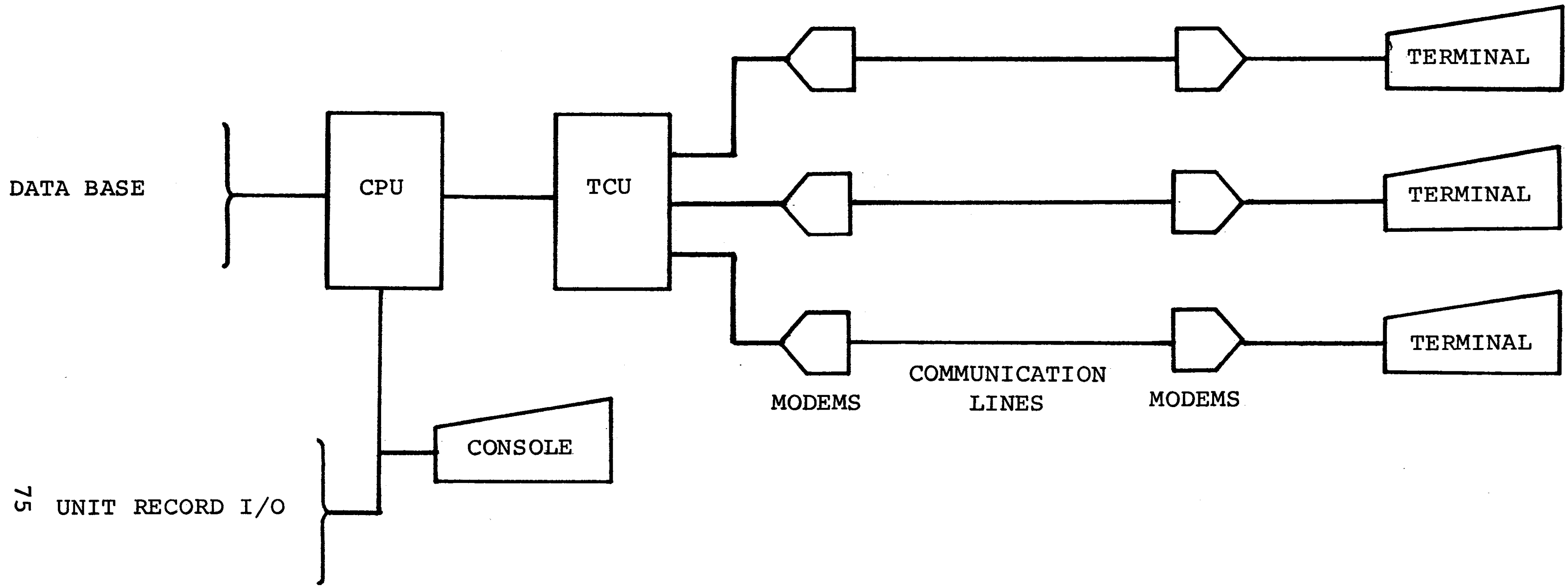


FIGURE 34. ACTUAL PRODUCTION SYSTEM.

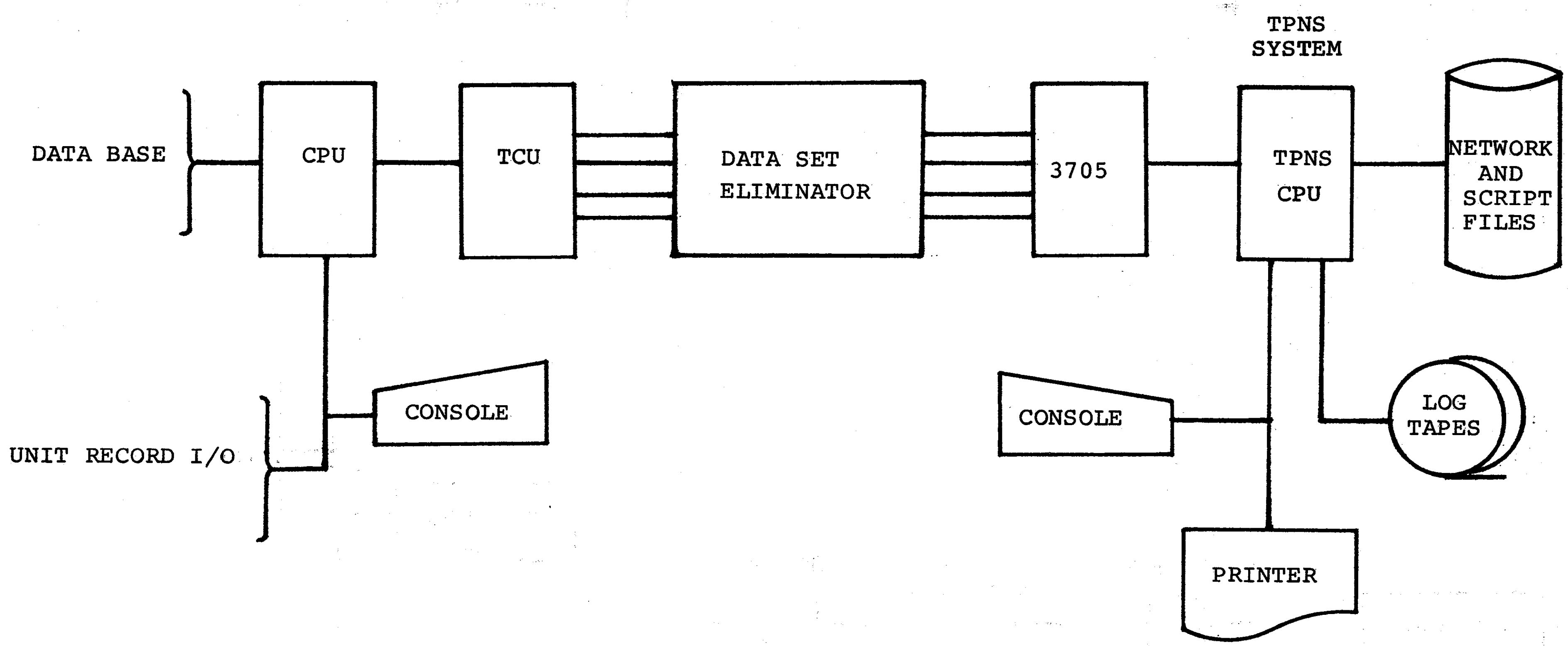


FIGURE 35. BENCHMARKING SYSTEM.

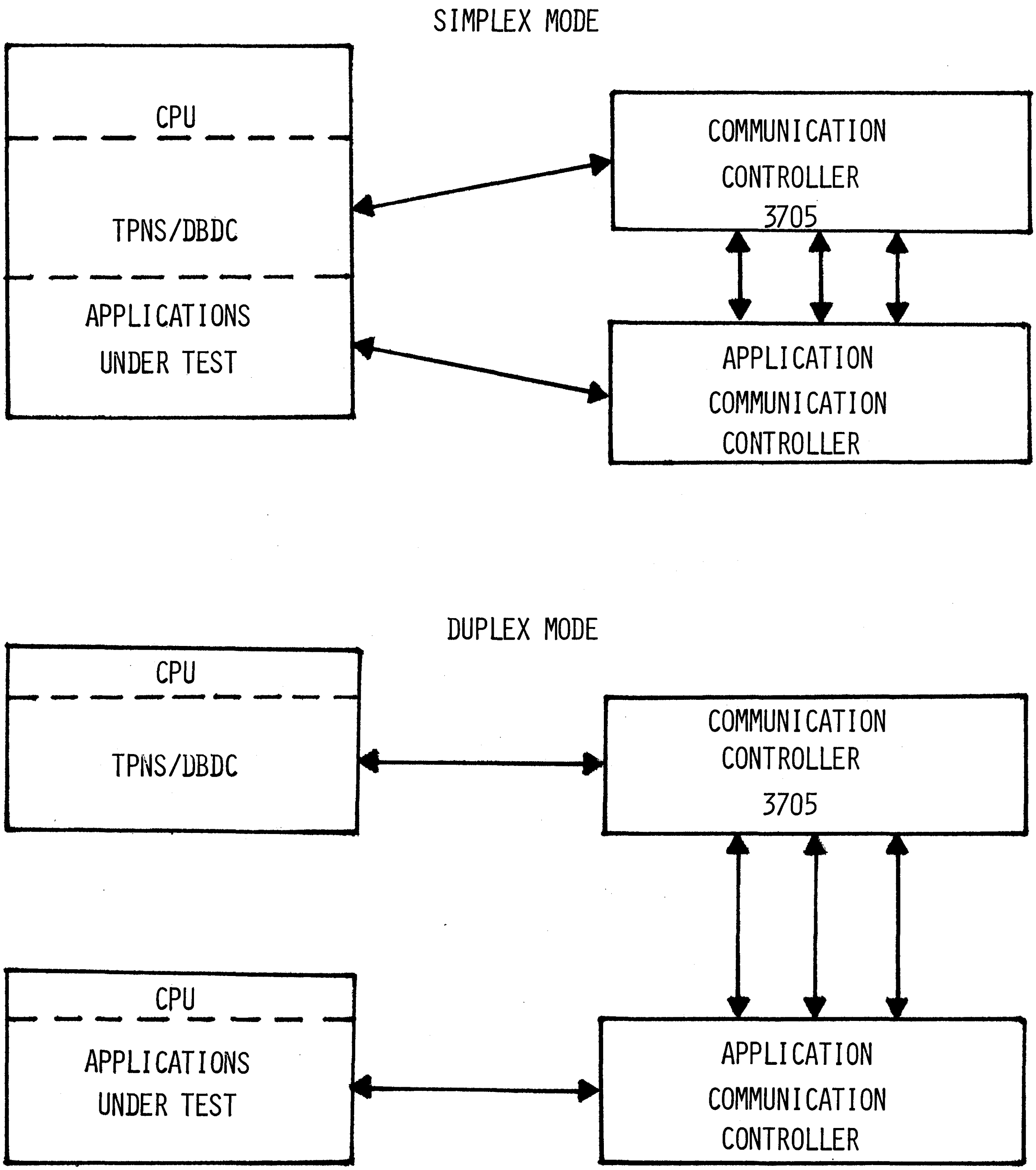


FIGURE 36. USE OF DRIVERS FOR PERFORMANCE PREDICTIONS.

processor fast enough (cycle time) such that it will not saturate before the processor under test. Obviously, in duplex mode the driver performance does not interfere with the application. Therefore, with the proper scripts, on-line performance tests may be adequately accomplished.

As noted earlier, operating the total system for performance analysis can be very costly. Although benchmarking is intended to reduce these costs, it can also be quite costly from both the supplier and the user point of view. Another approach to system analysis is synthetic benchmarking. This involves the creation of "model installations" to represent real customer workloads which might consist of TSO and IMS, with their associated TP network activity and batch workload. Multi-user mixed workloads make benchmarking virtually impossible in some cases, therefore, emphasis has been placed on synthetic benchmarking for large data base applications. Special attention has been given to the usability of drivers for data base and data communication workload activity. A program is used to characterize the transaction types and the TP network. Output from these programs would be scripting code to drive synthetic message processing programs and to simulate the TP network with its traffic.

The synthetic system might be a model of the system shown in Figure 35. This approach would reduce benchmarking costs as well as add flexibility. For example, in a normal benchmark repetition of certain runs would require the data base to be reset. This would be a simple task for the model installation. The accuracy of results is reduced as one moves one step further from actual production operation.

5.0 Summary

Capacity planning is presented as a performance oriented approach to managing computer resources where the primary driving forces for planning as well as total system capacity depend upon the user service objectives. A major emphasis is placed on organizing and understanding the current data processing environment. Through quantification and analysis of the present environment, it is possible to define future requirements. Recognizing that capacity planning is an art, analysis and prediction is based upon empirical data, guidelines, rules of thumb and experience. Therefore, tracking performance parameters on a continuing basis is paramount to the processs. This procedure is intended to gain certain system insights not always possible from a blitz type of data gathering effort. From a continuous monitoring and analysis process, when resource requirements are projected and assessed, it is possible to develop empirical relationships, guidelines and rules of thumb.

One of the major points to be understood from a data collection point of view, is the necessity of an organized structure in which to analyze measured data. The structure for analysis is developed around the major application areas (marketing, financial, accounting, etc.) or departments (engineering, sales, data processing, etc.) which use the computing facilities. For example, this means that the use of resources as measured by utilization values will be segmented and accounted for by application area. Analysis for capacity planning requires this type of segmentation because current as well as future workloads are defined by application area. Therefore, if a department is able to understand its current data processing needs and the future requirements are projected in the same terms, the planning process becomes more manageable.

The term capacity planning connotes detail modelling (queueing, GPSS, CSS, etc.) for analysis and prediction. As pointed out in this bulletin, there is a definite place in the growth of the capacity planning process for these techniques but it is possible to do meaningful analysis and forecasting without referencing a queueing relationship or discrete simulator. There are measurement tools available:

- MF/1 - RMF
- SMF
- CICS Performance Analyzer
- IMS/VS Report Print Program.

through which the current system may be understood. Then, by data trends and historical data (e.g., 10-3350's reduces TSO response time by 0.5 of a second), meaningful future resource requirements may be defined. The primary message is that capacity planning may be adequately initiated purely from an empirical or a measurement and feedback type of environment.

When the current environment is sufficiently understood and confidence is established in critical modelling parameters (workload levels, resource service requirements by application, etc.), then an installation may move to single server queueing analysis or certain automated predictive tools which will enhance projections. There are a tremendous number of models and techniques available for analyzing computer systems but the crucial factors are the understanding of the system operation (hardware and software) and the accuracy of the data which characterizes this operation. Therefore, detail modelling techniques should be introduced into the capacity planning effort only after achieving confidence in system interpretation and data accuracy.

It will take time for capacity planning to provide the kind of insight and understanding necessary for highly accurate forecasting and prediction. However, these initial efforts will make greater accuracy possible in the future as more installations begin to track performance and report upon their findings. This will mean rules or thumb will become guidelines, guidelines will become empirical relationships and empirical relationships will become laws.

6.0 References

1. "Analysis of some queuing models in real-time systems", IBM manual number F20-0007-1, published by IBM, Mechanicsburg, Pa. 1971.

2. Kendall, D. G., "Stochastic Processes Occurring in the Theory of queues And Their Analysis By The Method Of The Imbedded Markoy Chain", Annals of Mathematical Statistics, Vol. 24, 338-354, 1953.

3. Martin, James, "Design of Real Time Computer Systems", Page 395, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1967.

4. Little, John D. C., "A Proof For The Queueing Formula N = LTr," The Operations Research Society of America, Vol. 9, No. 3, 1961.

APPENDIX - A

Capacity Planning Presentation

During 1976, a capacity planning presentation based on the documentation contained in this appendix was presented to many executives (customer and IBM), marketing teams, GUIDE, SHARE and many internal IBM meetings. The presentation was used to introduce the methodology and techniques described in this bulletin. This bulletin will not preclude the necessity of presenting capacity planning, but certainly should enhance such a presentation. Now, the audience will have available a written narrative to support the methodology. This presentation is provided to be used directly or as aid for developing your own. This bulletin should replace the need for many of these presentations.

A

CAPACITY PLANNING

METHODOLOGY

DP MANAGEMENT WANT ANSWERS.....

- "WHAT HAPPENS TO MY ONLINE SYSTEM WHEN I ADD 150 TERMINALS TO IT NEXT YEAR?"
- "CAN I INSTALL IMS/VS AND MAINTAIN CURRENT SYSTEM PERFORMANCE LEVELS?"
- "WHEN SHOULD I UPGRADE MY CPU?"
- "HOW MUCH MEMORY DO I NEED TO RUN IMS/VS, TSO, AND BATCH?"
- "AT WHAT LEVEL OF PERFORMANCE IS MY SYSTEM RUNNING TODAY?"
- "HOW MUCH CAPACITY DO I HAVE LEFT WHEN I AM RUNNING MY CPU AT 100% UTILIZATION?"

OUR SOLUTION IS CAPACITY PLANNING.....

AGENDA

- 0 CAPACITY PLANNING OVERVIEW
  - o BASIC DEFINITIONS
  - o RESOURCE CAPACITY
    - o WORKLOAD
    - o AVAILABLE CAPACITY
  - o SYSTEM CAPACITY
    - o WORKLOAD
    - o RESOURCE CAPACITY
    - o USER SERVICE OBJECTIVES
  - o UNATTAINABLE CAPACITY
- 0 DATA COLLECTION AND REDUCTION
  - o PERFORMANCE MEASUREMENT TOOLS
  - o OPERATING SYSTEM DATA REQUIREMENTS
  - o SUBSYSTEM DATA REQUIREMENTS
  - o PERFORMANCE DATA USAGE
- 0 CAPACITY PLANNING IMPLEMENTATION
  - o MANAGEMENT AND CONTROL OF DP APPLICATION AREAS
  - o PHASED APPROACH
  - o EXAMPLE
  - o PERFORMANCE PREDICTION
- 0 CONCLUSIONS

CAPACITY PLANNING

OVERVIEW

CAPACITY PLANNING

O ORGANIZING STRUCTURE FOR SYSTEM ANALYSIS AND UNDERSTANDING

O PERFORMANCE ORIENTED APPROACH TO COMPUTER FACILITY MANAGEMENT

O MONITORS THE UTILIZATION OF SYSTEM RESOURCES

O MANAGEMENT FOR BEST OVERALL USER SATISFACTION

O INTEGRAL ON-GOING PART OF COMPUTER MANAGEMENT FUNCTION

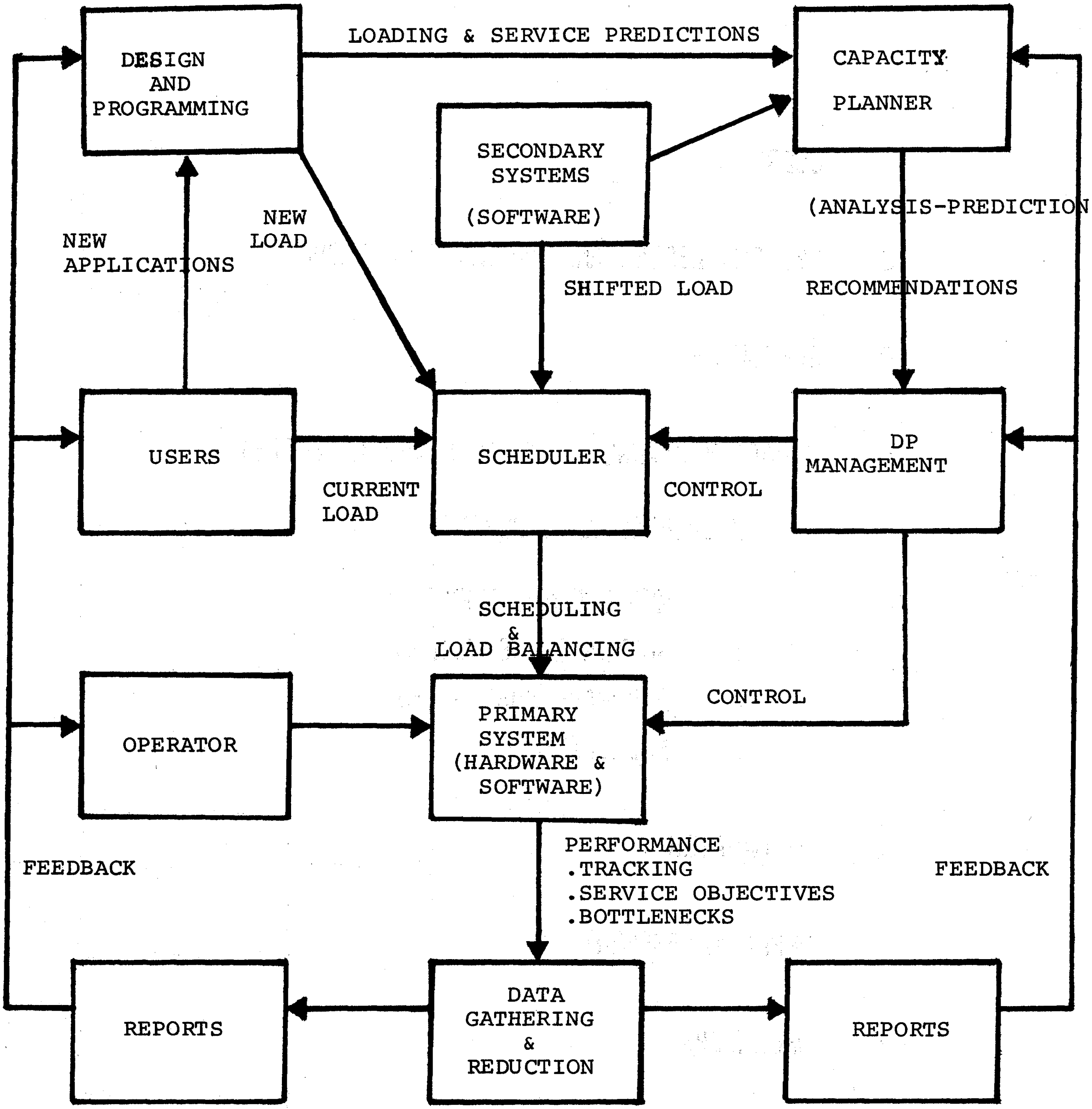
CAPACITY PLANNING - BASIC FUNCTIONS

- ESTABLISH/TRACK PERFORMANCE LEVELS
  - WORKLOAD LEVELS
  - RESOURCE SERVICE LEVELS (UTILIZATION)
  - SYSTEM PARAMETERS (PAGING, WAITING, ETC.)
- IDENTIFY SYSTEM BOTTLENECKS
- ESTABLISH/TRACK USER SERVICE OBJECTIVES
- PERFORM:
  - SCHEDULING
  - SCHEDULE TRACKING
  - LOAD BALANCING
- PREDICTION/FORECASTING
  - WORKLOAD/SERVICE REQUIREMENTS
  - OVERALL PERFORMANCE IMPACT
- PARAMETER AND LOAD SELECTION
  - ANALYTICAL MODELS (QUEUEING)
  - DISCRETE SIMULATION (GPSS, CSS)
  - SIMULATION DRIVERS
- REPORTING
  - PERTINENT DATA
  - TYPES OF DISPLAYS
  - RECIPIENTS
- SYSTEM RECOMMENDATIONS
  - UPGRADING
  - LOADING
  - REMOVAL

PRIMARY PEOPLE INVOLVED

- 0 USERS
- 0 SYSTEMS DESIGN AND PROGRAMMING GROUP
- 0 SCHEDULER
- 0 CAPACITY PLANNER (MAY BE SEVERAL PEOPLE)
  - o KNOWLEDGE REQUIREMENTS
    - o MEASUREMENT TOOLS
    - o SOFTWARE SUBSYSTEMS
    - o HARDWARE SUBSYSTEMS
    - o MODELLING
- 0 DP MANAGER
- 0 UPPER MANAGEMENT
- 0 OPERATORS

CAPACITY PLANNING FLOW



NOTE: ALTHOUGH, MOVEMENT THROUGH STRUCTURE SHOULD BEGIN AT THE LOADING POINT (CURRENT, SHIFTED AND NEW) INPUT TO THE SCHEDULER, THE OVERALL CONCEPT IS THAT OF A CONTINUOUS FLOW.

BASIC SOFTWARE

- O OPERATING SYSTEMS
  - o MVS
  - o SVS
  - o VS1
  - o VM
- O SUBSYSTEMS
  - o TSO
  - o APL
  - o VSPC
  - o IMS
  - o CICS
  - o BATCH

BASIC HARDWARE

0 CPU (UP, AP, MP)

0 CHANNELS

0 CONTROL UNIT

0 DASD/TAPES

0 PRINTERS

0 COMMUNICATION CONTROLLERS

0 LINES

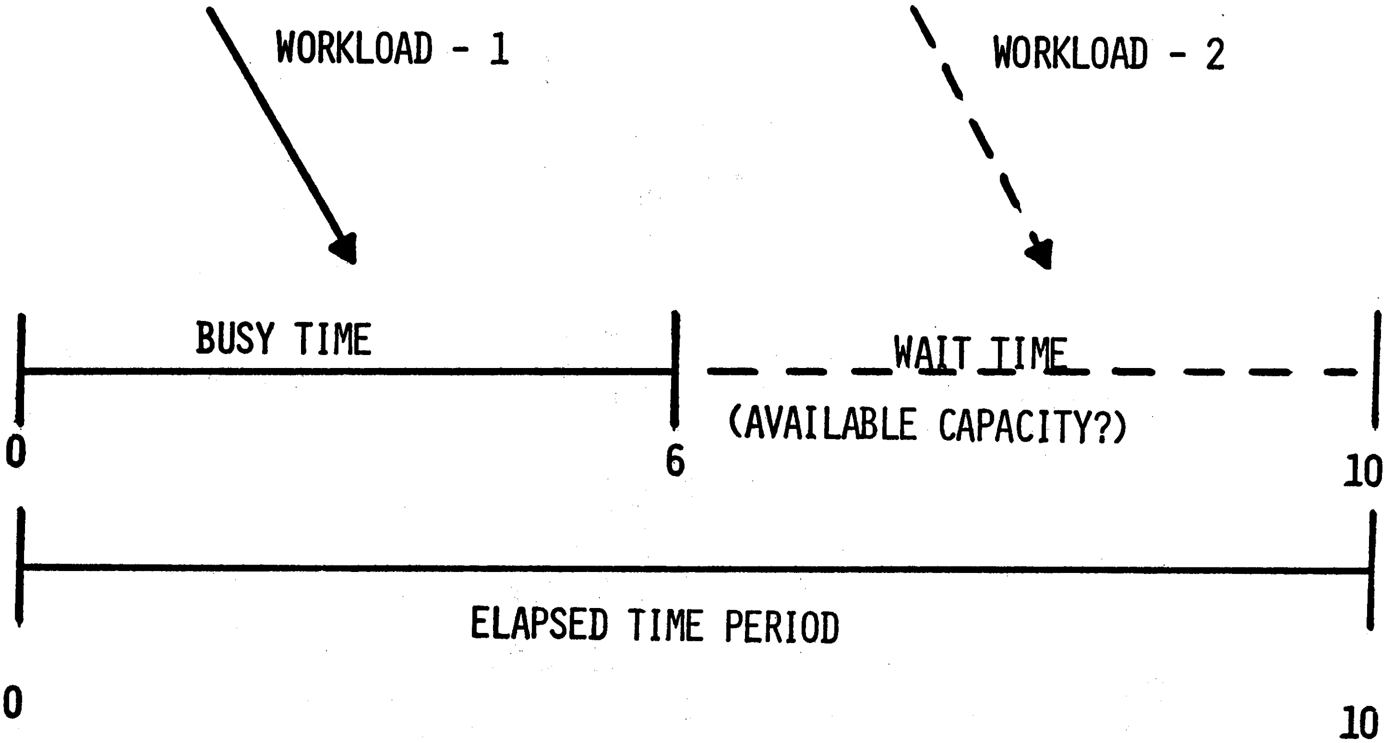
0 CLUSTER CONTROLLERS

0 CONTROLLER ADAPTERS

0 AUXILIARY STORAGE

0 TERMINALS

RESOURCE CAPACITY AND ITS INTERACTIONS

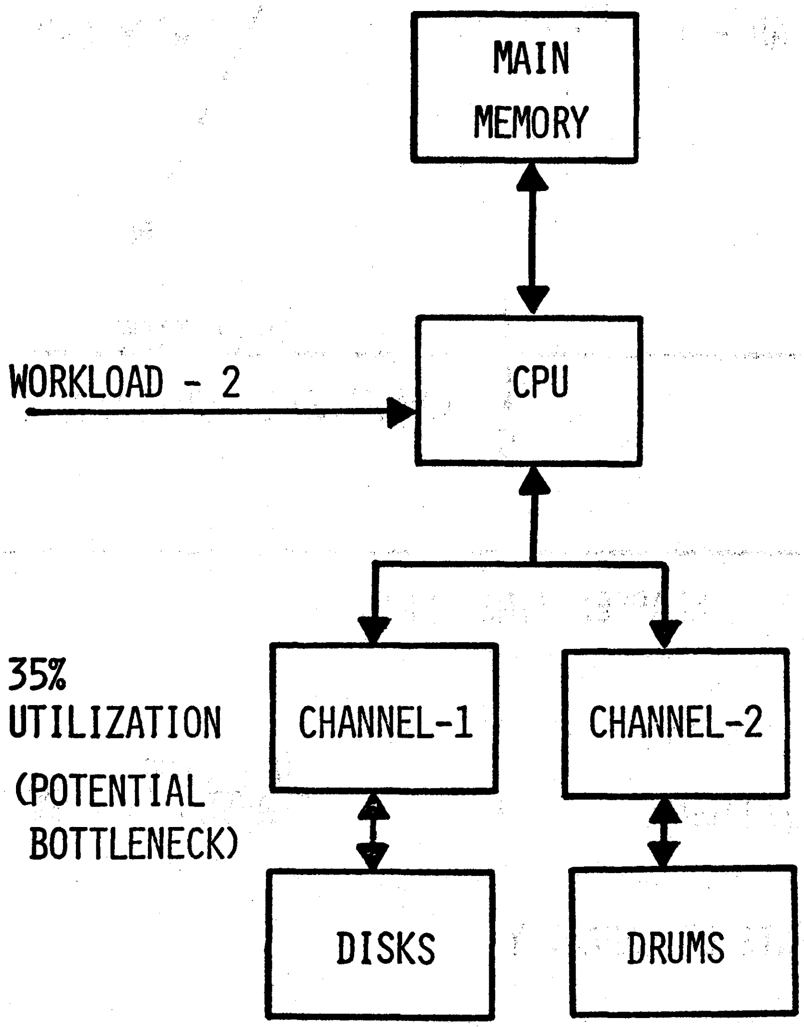


$$\text{UTILIZATION} = \frac{\text{BUSY TIME}}{\text{ELAPSED TIME PERIOD}}$$

ELEMENTS OF CAPACITY

- 0 BUSY TIME
- 0 WAIT TIME
- 0 ELAPSED TIME PERIOD
- 0 UTILIZATION
- 0 WORKLOAD

SUBSYSTEM INTERACTION



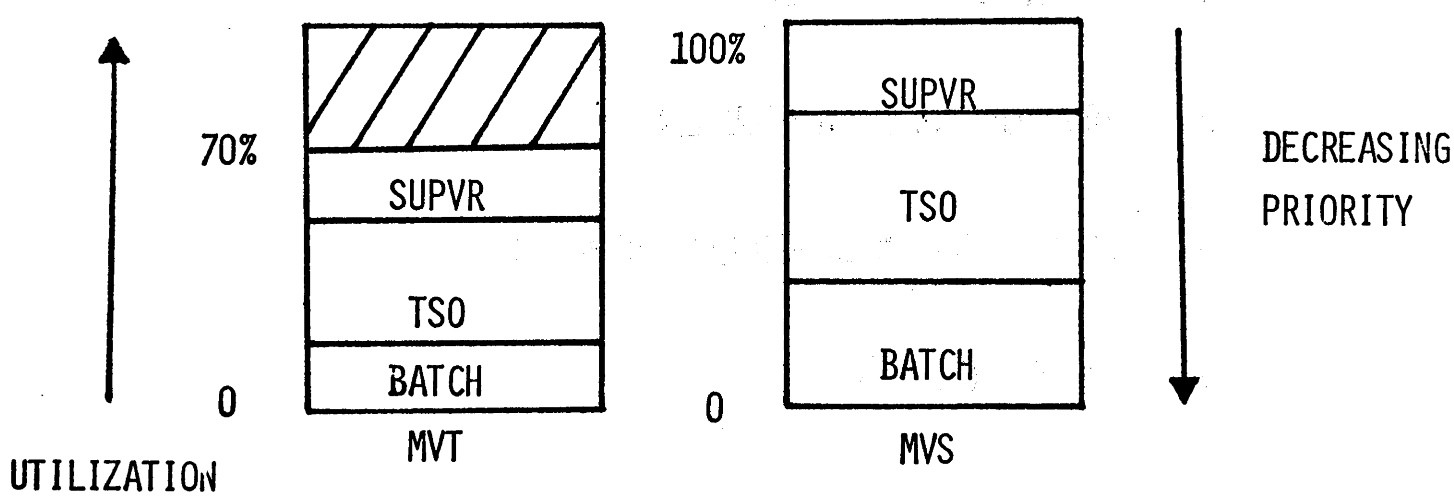
AVAILABLE CAPACITY IS VERY DEPENDENT UPON WORKLOAD, ITS REQUIRED RESOURCES AND THEIR INTERACTION.

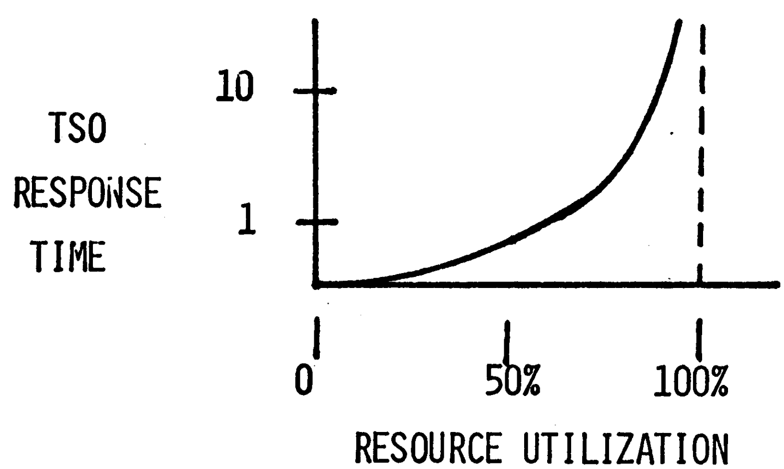
SYSTEM CAPACITY

0 ELEMENTS OF CAPACITY

- o WORKLOAD
  - o BATCH (INITIATORS)
  - o ON-LINE (TERMINALS)
- o RESOURCE UTILIZATION
- o <u>USER SERVICE</u>*
  - o RESPONSE TIME
  - o TURNAROUND TIME

0 CAPACITY CONSIDERATIONS (CPU)





* MOST CRITICAL INDICATOR OF SYSTEM CAPACITY

UNATTAINABLE CAPACITY

- O SYSTEM DOWN TIME
  - o MAINTENANCE
  - o WORK IN PROCESS
  - o START-UP TIME
- O OPERATIONAL PROBLEMS
- O PROGRAM AND DATA PROBLEMS
- O VARIATIONS IN SCHEDULING DEMANDS
- O SYSTEM RECOVERY

UNATTAINABLE CAPACITY

EXAMPLE: REFERENCE COMPUTER WORLD ARTICLE ON CHASE MANHATTAN BANK ENTITLED "100% UTILIZATION, IMPOSSIBLE DREAM", DATED FEBRUARY 19, 1975.

FACTORS ACTING TO REDUCE EFFECTIVE CPU CAPACITY:

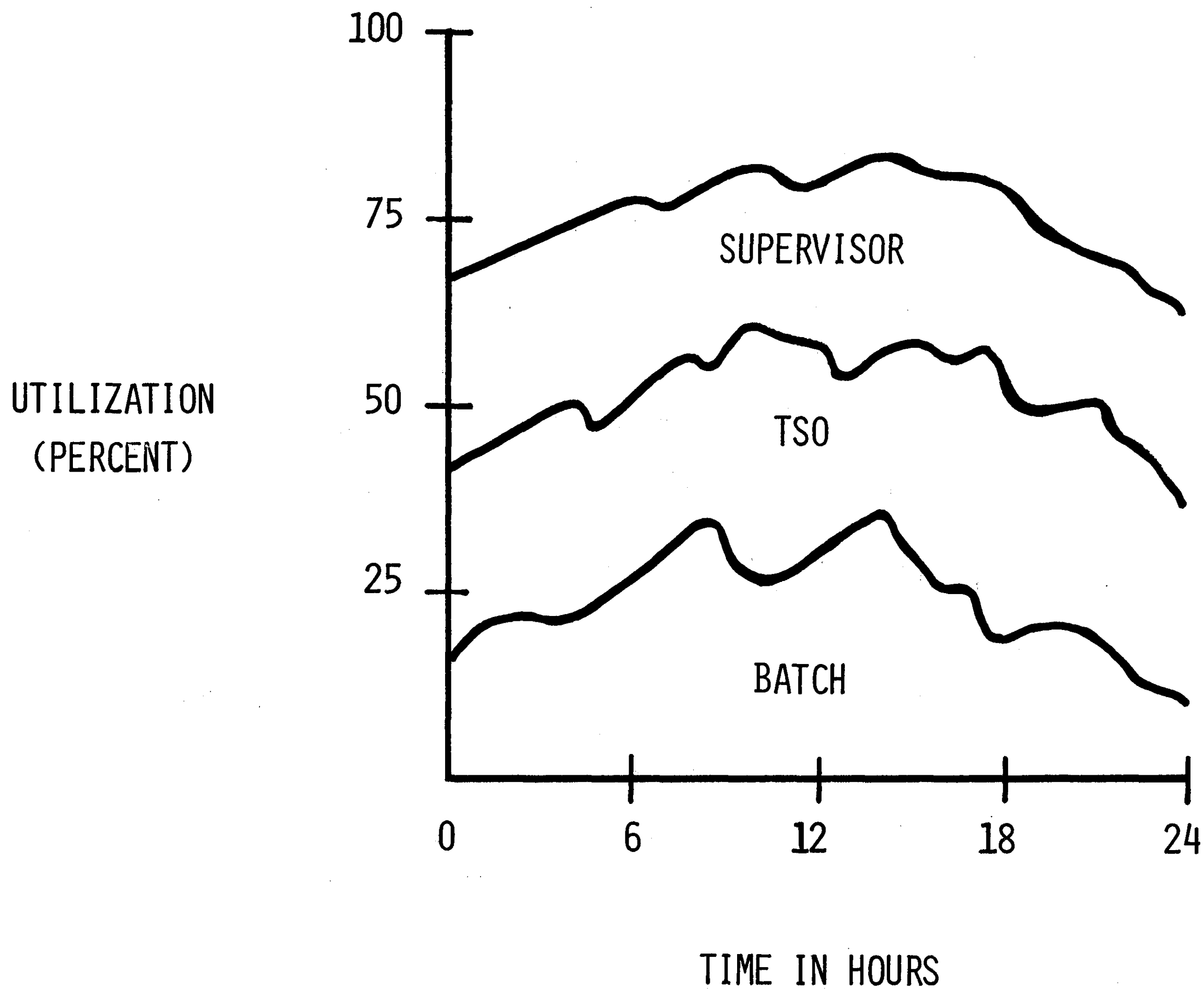
1. OPERATIONAL INEFFECTIVENESS (9%)
   A. SYSTEM DOWNTIME
   B. OPERATIONAL PROBLEMS
   C. PROGRAM AND DATA PROBLEMS
2. UNREACHABLE CAPACITY (15%)
   A. DIFFERENCE IN SYSTEM REQUIREMENTS
3. RECOVERY (5%)
   A. PREDECESSOR-FEEDER RELATIONSHIPS
4. OPERATIONAL INEFFECTIVENESS/ARRIVAL OF SCHEDULED DEMANDS (7%)

TOTAL		36%
THRESHOLD CAPACITY	100 - 36 =	64%

DATA

COLLECTION & REDUCTION

EXAMPLE CHART
(CPU UTILIZATION VS TIME)



PERFORMANCE MEASUREMENT TOOLS

O COLLECTORS

C1 - HARDWARE MONITOR

C2 - SMF (SYSTEM MANAGEMENT FACILITY)

C3 - GTF (GENERAL TRACE FACILITY)

C4 - TS TRACE (TIME SHARING TRACE)

C5 - IMS/VS SYSTEM LOG

O ANALYZERS

A1 - HARDWARE MONITOR REPORT PROGRAM

A2 - SGP (STATISTICS GENERATING PACKAGE)

A3 - SMF GRAPHICAL ANALYZER

A4 - CAPACITY MANAGEMENT AID

A5 - IMS/VS LOG TRANSACTION ANALYSIS

A6 - IMS/VS STATISTICAL ANALYSIS

O COLLECTOR - ANALYZER

CA1 - MF/1 (MEASUREMENT FACILITY 1)

CA2 - RMF (RESOURCE MEASUREMENT FACILITY)

CA3 - SVSPT (VS2 PERFORMANCE TOOL)

CA4 - VS1PT (VS1 PERFORMANCE TOOL)

CA5 - SIR (SYSTEM INFORMATION ROUTINE)

CA6 - CICS PERFORMANCE ANALYZER II

CA7 - CICS PLOT

CA8 - CICS DYNAMIC MAP

CA9 - IMS/VS MONITOR REPORT PRINT PROGRAM

CA10 - IMS/TRAPDL1

CA11 - APL SYSTEM

CA12 - UTILITY IEHLIST (LIST VTOC)

CA13 - MVS/SVS SYSTEM AND JOB IMPACT ANALYSIS

USE OF PERFORMANCE MEASUREMENT TOOLS

MEASUREMENT TOOL	TYPE	MVS	SVS	VS1
1. HARDWARE MONITOR		X	X	X
2. SMF	SCP	X	X	X
3. GTF	SCP	X	X	X
4. TS TRACE	SCP	NA	X	NA
5. IMS/VS SYSTEM LOG	PP	X	X	X
6. HARDWARE MONITOR RPT. PGM.		X	X	X
7. SMF GRAPHICAL ANALYZER	FDP	NA	X	X
8. CAPACITY MANAGEMENT AID	FDP	X	X	X
9. SGP - STATISTICS GATHERING PKG.	FDP	X	X	X
10. IMS/VS LOG TRANSACTION ANALYZIS	PP	X	X	X
11. IMS/VS STATISTICAL ANALYSIS	PP	X	X	X
12. MF/1	SCP	X	NA	NA
13. RMF	PP	X	NA	NA
14. VS2PT	IUP	NA	X	NA
15. VS1PT	IUP	NA	NA	X
16. SIR - SYSTEM INFO. ROUTINE	IUP	X	NA	NA
17. CICS PERFORMANCE ANALYZER II	FDP	X	X	X
18. CICS PLOT	FDP	X	X	X
19. CICS DYNAMIC MAP	FDP	X	X	X
20. IMS/VS MONITOR REPORT PRINT PROGRAM	PP	X	X	X
21. IMS/TRAPDL1	FDP	X	X	X
22. APL SYSTEM	PP	X	X	X
23. UTILITY IEHLIST (LIST VTOC)	PP	X	X	X
24. MVS/SVS SYSTEM AND JOB IMPACT ANALYSIS	IUP	X	X	NA

OPERATING SYSTEMS

	MVS	SVS	VS1
HARDWARE MONITOR	X	X	X
SMF	X	X	X
SGP	X	X	X
GTF	X	X	X
MF1- RMF	X		
SVSPT		X	
VS1PT			X
SIR	X		
IMPACT ANALYZER	X	X	

PERFORMANCE PARAMETERS

PARAMETERS \ MEASUREMENT TOOL	HDW. MON.	SMF	SGP	MF/1	RMF	SVSPT	VS1PT	SIR	GTF	IMPACT ANAL.			
CPU													
. TOTAL WAIT	X	X	X	X	X	X	X						
. IDLE WAIT	X	(X)	(X)			X							
. I/O WAIT	X					X	X						
. PAGE WAIT						(X)	X						
. UTILIZATION	X	X	X	X	X	X	X	(X)		X			
. CPU TIME (PROBLEM PROGRAM)	X	(X)	(X)					(X)					
. CPU TIME (SUPERVISOR)	X							(X)		X			
. SYSTEM PAGING RATE	(X)			X	X		X		X	X			
. USER PAGING RATE	(X)	(X)	(X)	X	X				X	X			
. TOTAL PAGING RATE	(X)			X	X	X	X	X	X	X			
. SWAPPING RATE		(X)	(X)	X	X	X							
. PAGES PER SWAP-OUT		(X)	(X)	X	X	(X)							
. PAGES PER SWAP-IN		(X)	(X)	X	X	(X)							
. CPU, CHANNEL OVERLAP	X			X	X	X	X						
. MULTIPROGRAMMING LEVEL BY TIME		(X)	(X)	(X)	(X)	X							
. NUMBER ACTIVE INITIATORS BY TIME		(X)	(X)				X	X					
MEMORY													
. AVAILABLE FRAMES BY TIME				X	X	X	X	X		(X)			
. WORKING SET SIZE BY USER		(X)	(X)			(X)	X	X		X			

NOTE: AN "X" IS AN INDICATION THAT PARAMETER IS COLLECTED DIRECTLY, WHEREAS "(X)" INDICATES FURTHER REDUCTION IS REQUIRED OR PARAMETER IS ONLY PARTIALLY COLLECTED.

PERFORMANCE PARAMETERS

PARAMETERS \ MEASUREMENT TOOL	HDW. MON.	SMF	SGP	MF/1	RMF	SVSPT	VS1PT	SIR	GTF	IMPACT ANAL.			
CHANNEL													
. UTILIZATION	X			X	X	X	X						
. TOTAL EXCP'S	X	(X)	(X)	(X)	(X)	(X)			X				
. BYTES/CHANNEL/TIME	X								X				
. CHANNEL QUEUE SIZE (RQE)						X							
I/O DEVICE													
. UTILIZATION				X	X	(X)	X						
. EXCP'S/DEVICE	X	(X)	(X)	(X)	(X)	X			X				
. EXCP'S/DEVICE/JOB									X				
. BYTES/DEVICE/TIME	X								X				
. DATA SET ACCESS RATE/JOB		X	X										
. DEVICE QUEUE SIZE				X	X	(X)							
. AVERAGE BYTES/EXCP	X												

NOTE: AN "X" IS AN INDICATION THAT PARAMETER IS COLLECTED DIRECTLY, WHEREAS "(X)" INDICATES FURTHER REDUCTION IS REQUIRED OR PARAMETER IS ONLY PARTIALLY COLLECTED.

BATCH CAPACITY PLANNING

MEASUREMENT TOOL / MEASURED DATA	C2	A2	A3	A4	CA5	CA12
LOADING						
O BY GROUPS TRACKED BY INSTALLATION						
o JOBS ARRIVING/HOUR	X	X				
o EXCP'S/CHANNEL	X	X				
o EXCP'S/DEVICE	X	X				
o BYTES/EXCP						X
O BY LARGE PRODUCTION JOBS						
o EXCP'S/CHANNEL	X	X				
o EXCP'S/DEVICE	X	X				
o BYTES/EXCP	X	X				X
NOTE: TOTAL RECORDS PROCESSED FILE ORGANIZATION & SIZE						
SERVICE						
O BY GROUPS TRACKED BY INSTALLATION						
o CPU TIME/JOB	X	X				
o ELAPSED TIME/JOB	X	X				
o JOBS COMPLETING/HOUR	X	X				
O BY HEAVY PRODUCTION JOBS						
o CPU TIME	X	X				
o ELAPSED TIME	X	X				
NOTE: NO RECORD OF JOBS STARTED FROM CONSOLE						
RESOURCE UTILIZATION						
o AVERAGE MEMORY WS SIZE BY CLASS					X	
o AVERAGE MEMORY WS SIZE BY LARGE JOB					X	
o GRAPHICAL ANALYSIS			X	X		

TSO CAPACITY PLANNING

MEASURED DATA \ MEASUREMENT TOOLS	C2	C3	C4	A2	CA1	CA5
LOADING						
o TRANSACTIONS/SECOND/CLASS		X	X		X	
o MEAN NUMBER CONCURRENT USERS/PERIOD	X			X		
o MEAN NUMBER SWAPS/TRANSACTION			X		X	
o AVERAGE PAGING LOAD/SWAP			X		X	
o TOTAL EXCP'S/DEVICE/PERIOD	X			X		
o TERMINAL I/O LOAD BY USER	X			X		
o CONNECT TIME	X			X		
SERVICE						
o AVERAGE CPU TIME/CLASS		X	X		X	
o TOTAL CPU TIME/PERIOD		X	X		X	
o TOTAL ELAPSED TIME PERIOD		X	X			
o AVERAGE USER RESPONSE TIME		X	X			
o AVERAGE THINK TIME		X	X			
o AVERAGE CPU TIME/SWAP		X	X			
RESOURCE UTILIZATION						
o AVERAGE MEMORY WS SIZE BY USER						X
o AVERAGE MEMORY WS SIZE (TOTAL)		X	X			

APL CAPACITY PLANNING

MEASURED DATA	MEASUREMENT TOOL CA11	C2	A2	CA5
LOADING				
o TRANSACTIONS/SECOND/CLASS	X			
o MEAN NUMBER CONCURRENT USERS/PERIOD	X			
o MEAN NUMBER SWAPS/TRANSACTIONS	X			
o TOTAL EXCP'S/DEVICE/PERIOD		X	X	
SERVICE				
o AVERAGE CPU TIME/CLASS	X			
o AVERAGE CPU TIME/SWAP (IN & OUT)	X			
o AVERAGE THINK TIME	X			
o TOTAL CPU TIME	X			
o AVERAGE USER RESPONSE TIME	X			
RESOURCE UTILIZATION				
o TOTAL WORKING SET SIZE				X
o WORK SPACE WS SIZE AT SWAP	X			

CICS CAPACITY PLANNING

MEASURED DATA / MEASUREMENT TOOL	C2	A2	CA6	CA7	CA8	CA12
LOADING						
o TRANSACTION TYPES			X			
o BYTES/TRANSACTION (IN & OUT)			X			
o TRANSACTIONS/SECOND/TERMINAL			X			
o LOGICAL FILE ACCESSES/TRANSACTION*			X			
o EXCP'S/DEVICE	X	X				
o TOTAL EXCP'S BY CHANNEL	X	X				
o BLOCK SIZE BY FILE						X
SERVICE						
o AVERAGE CPU TIME/TRANSACTION			X			
o ELAPSED TIME/TRANSACTION			X			
o TOTAL CPU TIME			X			
o TOTAL ELAPSED TIME			X			
RESOURCE UTILIZATION						
o SHORT-ON-STORAGE			X			
o MAXIMUM TASKS			X			
o STORAGE UTILIZATION				X	X	

* MUST IDENTIFY ACCESS METHOD USED

IMS CAPACITY PLANNING

MEASUREMENT TOOL MEASURED DATA	C2	A2	C5	A5	A6	CA9	CA10	CA12
LOADING								
o TRANSACTION TYPES			X		X			
o BYTES/TRANSACTION			X		X			
o TRANSACTIONS/SECOND/TERMINAL			X		X			
o TRANSACTIONS/SECOND/LINE			X		X			
o LOGICAL FILE ACCESSES/TRANSACTION						X	X	
o PHYSICAL FILE ACCESS/MPP (BATCH)	X	X				X		
o EXCP'S/DEVICE	X	X						
o TOTAL EXCP'S BY CHANNEL	X	X						
o BLOCK SIZE BY FILE								X
o TRANSACTIONS BY MPP					X			
o MPP BY ADDRESS SPACE						X		
o NUMBER OF MFS PREFETCH I/O'S						X		
o NUMBER OF MFS IMMEDIATE FETCH I/O'S						X		
o NUMBER OF MFS DIRECTORY I/O'S						X		
o NUMBER OF I/O'S DUE TO INSUFFICIENT MESSAGE QUEUES						X		
o TOTAL NUMBER OF TRANSACTIONS					X		X	
o TOTAL NUMBER LOG RECORDS								

IMS CAPACITY PLANNING

MEASURED DATA / MEASUREMENT TOOL	C2	A2	C5	A5	A6	CA9	CA10	CA12
SERVICE								
o TOTAL CPU TIME/TRANSACTION			X	X				
o CPU TIME/TRANSACTION (MPP)						X		
o TOTAL ELAPSED TIME/TRANSACTION			X	X				
o ELAPSED TIME/TRANSACTION (MPP)						X	X	
o SCHEDULE TO 1ST DL/1 CALL								
o CPU TIME						X		
o ELAPSED TIME						X		
o ELAPSED TIME (BATCH DL/1)							X	
RESOURCE UTILIZATION								
o MAXIMUM MESSAGE QUEUE SIZE						X		
o UNAVAILABLE BUFFER POOL SPACE						X		
o PROGRAM DEADLOCK OCCURANCES						X		

PERFORMANCE DATA USAGE

- 0 ESTABLISH CURRENT STATE OF SYSTEM
- 0 ESTABLISH TUNING LEVEL
- 0 MODEL DEVELOPMENT (PREDICTIONS/FORECASTS)
  - o HAND CALCULATIONS
  - o AUTOMATED
- 0 MODEL VALIDATION
- 0 PERFORMANCE TRACKING
  - o TUNING
  - o PROJECTIONS (PREDICTIONS/FORECASTS)
  - o SCHEDULING

PERFORMANCE DATA USAGE

PERFORMANCE DATA	SYSTEM STATE	TUNING LEVEL	MODEL VALIDATION	TRACKING		
				TUNING	PROJECTIONS	SCHEDULING
LOADING (BATCH, ON-LINE)	X	X	X		X	
RESOURCE UTILIZATIONS	X		X	X	X	
RESPONSE TIMES	X		X		X	
TURNAROUND TIMES	X		X		X	
AVAILABLE FRAMES COUNT		X		X		
PAGING RATES (IN/OUT)		X		X		
SWAPPING RATES (IN/OUT)		X		X		
NUMBER OF INITIATORS	X					
AVERAGE NUMBER OF ACTIVE INITIATORS	X		X			
CPU, CHANNEL OVERLAP		X		X		
JOB START TIMES						X
JOB COMPLETION TIMES						X
RESOURCE UTILIZATION BY JOB						X
THROUGHPUT						X
JOB PRINT START TIMES						X
JOB PRINT COMPLETION TIMES						X
LINES PRINTED (BY JOB, TOTAL)						X
DASD SEEK ANALYSIS		X				
DASD CONTENTION ANALYSIS		X				
SVC ANALYSIS		X				
BUFFER ANALYSIS		X				
VS ADDRESS ANALYSIS		X				
STORAGE UTILIZATION ANALYSIS		X				

VALIDATION OF MEASURING TOOLS

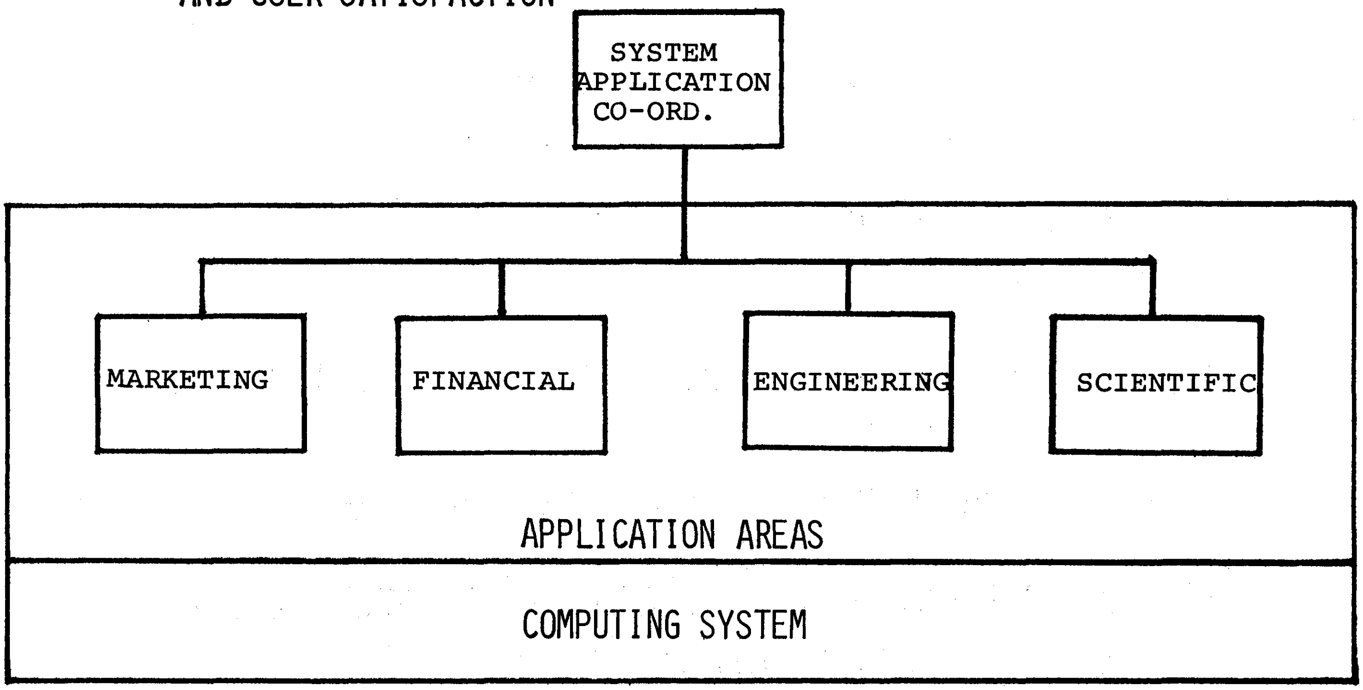
- O REGULAR CALIBRATION SCHEDULE USING STD JOBS
- O CALIBRATION FOR SYSTEM CHANGES (HARDWARE AND SOFTWARE)
- O DIFFERENT TOOLS COLLECTING SAME DATA

CAPACITY PLANNING

IMPLEMENTATION

CAPACITY PLANNING PROBLEM

MANAGEMENT AND CONTROL OF TOTAL SYSTEM WORKLOAD (CURRENT AND FUTURE) ON AN APPLICATION BASIS FOR BEST RESOURCE UTILIZATION AND USER SATISFACTION



- TOTAL WORKLOAD, COMBINATION OF WORKLOAD GENERATED BY EACH APPLICATION AREA
- WORKLOAD TYPES
  - BATCH: JOBS PER HOUR
  - TSO, APL: COMMANDS PER SECOND (CONCURRENT TERMINAL USERS)
  - CICS, IMS: TRANSACTION TYPES PER SECOND
- RESOURCE SERVICE REQUIREMENTS
  - PERCENT UTILIZATION
- USER SERVICE OBJECTIVES
  - RESPONSE TIME
  - TURNAROUND TIME

MANAGEMENT AND CONTROL OF DP APPLICATIONS
(GROWTH OF SYSTEM COMPLEXITY)

- IDENTIFY MAJOR APPLICATION AREAS
  - SALES
  - ACCOUNTING
  - ETC.
- ESTABLISH CURRENT WORKLOAD OF EACH APPLICATION AREA
  - JOBS/HOUR
  - TRANSACTIONS/SECOND - COMMANDS/SECOND
- ESTABLISH CURRENT CPU SERVICE REQUIREMENTS OF EACH APPLICATION BY PERIOD (DAY, WEEK, MONTH)
  - SMF JOB-HOURS
- IDENTIFY CRITICAL SERVICE REQUIREMENTS BY APPLICATION AREA, BY SHIFT
  - RESPONSE TIMES
  - TURNAROUND TIMES
- IDENTIFY CRITICAL INTERFACES
  - LOAD BALANCING
    - DATA SET PLACEMENT
    - SCHEDULING
- PROJECT ANTICIPATED WORKLOADS FOR EACH APPLICATION AREA
  - INCREASE/DECREASE
  - NEW PROJECTS

CAPACITY PLANNING IMPLEMENTATION (PHASES I - IV)

I. ESTABLISH CURRENT STATE OF SYSTEM

- O UNDERSTAND USERS CURRENT MEASURING TECHNIQUES AND PARAMETERS BEING MEASURED.
- O ESTABLISH/MONITOR WORKLOAD LEVELS
  - o BATCH (JOBS/MIN, JOB/CLASS/MIN)
  - o IMS, CICS (TRANSACTIONS/SECOND)
  - o TSO, APL (COMMANDS/SECOND)
  - o IDENTIFY HEAVY USERS
- O ESTABLISH/MONITOR CURRENT SERVICE LEVELS
  - o BATCH (ELAPSED TIME, THROUGHPUT)
  - o IMS, CICS, TSO, APL (RESPONSE TIME)
  - o UTILIZATION (BATCH, ON-LINE)
- O IDENTIFY REQUIRED REPORTS
- O ESTABLISH LEVEL OF UNATTAINABLE CAPACITY

NOTE: TRACKING (CONTINUOUS, SAMPLING)

CAPACITY PLANNING IMPLEMENTATION

II. ESTABLISH/MONITOR USER SERVICE OBJECTIVES

- 0 THE USER FOR TECHNICAL, ECONOMICAL, OR POLITICAL REASONS MAY BE REQUIRED TO ESTABLISH NEW RESPONSE TIME (ON-LINE) AND/OR TURNAROUND TIME (BATCH) REQUIREMENTS.

III. ESTABLISH/TRACK SCHEDULING SCHEME

- 0 UNDERSTAND USERS CURRENT SCHEDULING SCHEME
- 0 INTEGRATE SCHEDULING PROFILES WITH SYSTEM PERFORMANCE PROFILES
- 0 IDENTIFY SYSTEM BOTTLENECKS AND FORMULATE CORRECTIVE ACTION
  - o LOADING TRADEOFFS
  - o PURCHASE NEW HARDWARE
  - o SOFTWARE UPDATES

IV. SYSTEMIZATION (FINAL PHASE OF PUTTING TOTAL CAPACITY PLANNING ARCHITECTURE IN PLACE)

- 0 OVERALL PERFORMANCE TRACKING
- 0 PREDICTION/FORECASTING
- 0 REPORTING ANALYSIS
  - o CURRENT
  - o HISTORICAL

INITIATION OF CAPACITY PLANNING PROGRAM

0 EXAMPLE SYSTEM
  - o MVS (BATCH, TSO, CICS)

0 MEASUREMENT TOOLS
  - o MF/1 - RMF
  - o SMF
  - o CICS PERFORMANCE ANALYZER

0 IDENTIFY MAJOR APPLICATION GROUPS
  - o SALES
    - o 9 JOBS (MAJOR CICS APPLICATION)
  - o ACCOUNTING
    - o 10 JOBS
  - o DATA PROCESSING
    - o TESTING (SOME TSO)
    - o OPERATIONS
    - o SYSTEMS PROGRAMMING (SOME TSO)
  - o OPERATING SYSTEM
  - o MISCELLANEOUS

<u>NOTE</u>: APPLICATION CPU RESOURCE CONSUMPTION MEASURED IN JOB-HOURS/TIME PERIOD

INITIATION OF CAPACITY PLANNING PROGRAM (CONT'D.)

- DATA TO BE COLLECTED BY SHIFT (PHYSICAL/LOGICAL)
  - LOADING (MEANINGFUL, ONLY IF ACCOMPANIED BY ADDITIONAL SERVICE DATA, LONG PROCESSING, SHORT PROCESSING, ETC.)
    - BATCH (JOB RATE BY APPLICATION GROUPING)
    - TSO (ENDED TRANSACTION, TGETS/TIME PERIOD)
    - CICS (TRANSACTION RATE BY TYPE)
    - CHANNELS (EXCP RATE)
    - DEVICE (EXCP RATE)
  - UTILIZATION (CPU TIME)
    - TOTAL (ELAPSED - WAIT)
    - BY APPLICATION GROUP (JOB-HOURS)
      - BATCH
      - TSO
      - CICS
  - UTILIZATION
    - CHANNELS
    - I/O DEVICES
  - BATCH TURNAROUND TIMES
  - CICS RESPONSE TIME
  - TSO RESPONSE TIME
  - DETERMINE PEAK PERIODS
  - IDENTIFY DATA HOLES

INITIATION OF CAPACITY PLANNING PROGRAM (CONT'D.)

- DATA ANALYSIS
  - VALIDATION
  - IDENTIFY AND ANALYZE TRENDS
  - CORRELATION
    - LOAD
    - UTILIZATION
    - RESPONSE/TURNAROUND TIME

- BEGIN TO MAKE GROSS PROJECTIONS FROM TREND DATA AND VALIDATE PROJECTIONS
  - LOAD INCREASES
  - REQUIRED SERVICE (CPU JOB-HOURS)
  - RESOURCE UTILIZATIONS

- IDENTIFY NEW PROJECTS PLANNED FOR IMPLEMENTATION DURING NEXT 24 MONTH PERIOD

- MAKE GROSS PROJECTIONS ON NEW APPLICATIONS
  - LOAD INCREASE
  - REQUIRED SERVICE (CPU JOB-HOURS)
  - RESOURCE UTILIZATIONS

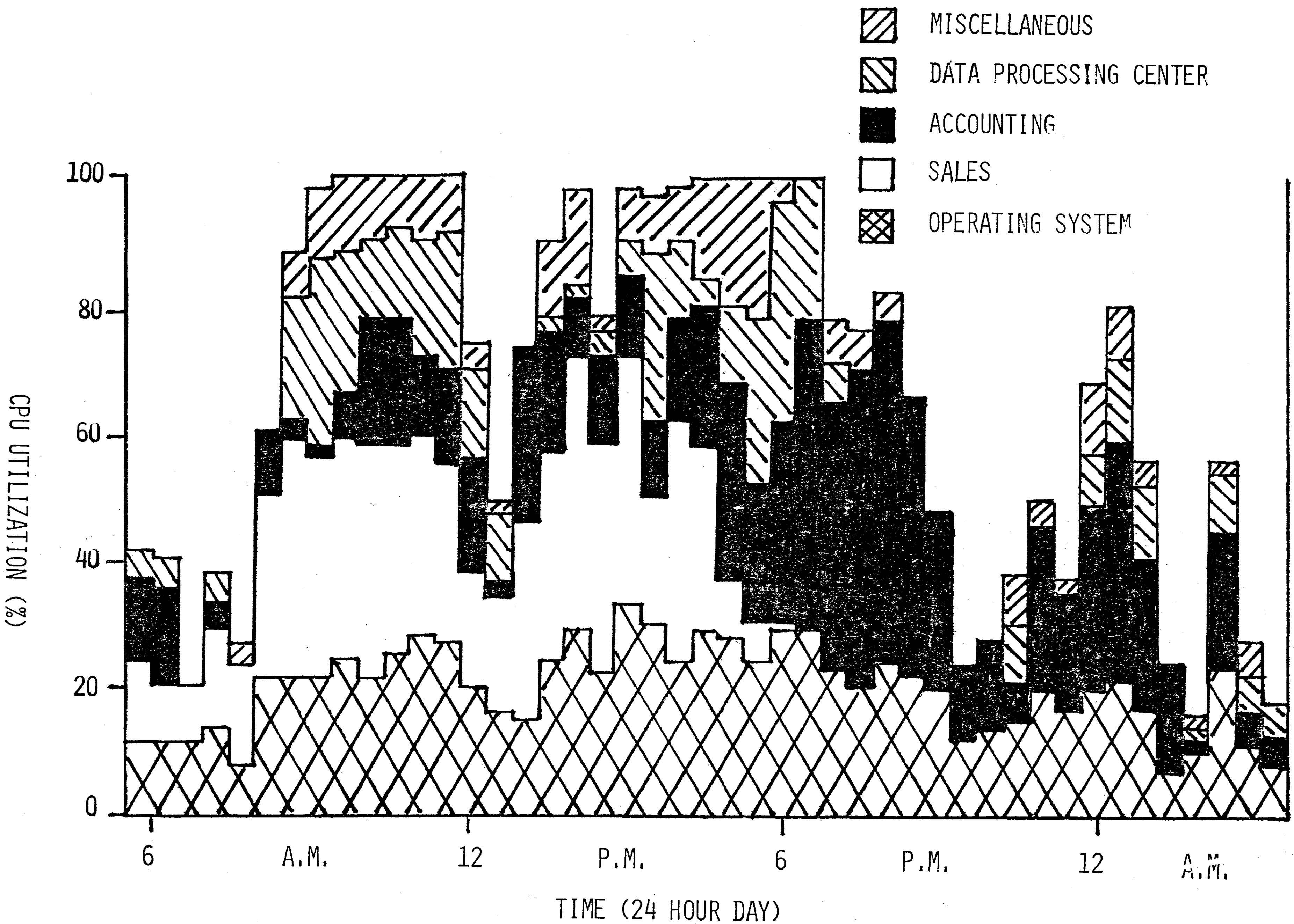
- BEGIN TO ESTABLISH GUIDELINES FOR PROJECTIONS

INITIATION OF CAPACITY PLANNING PROGRAM (CONT'D.)

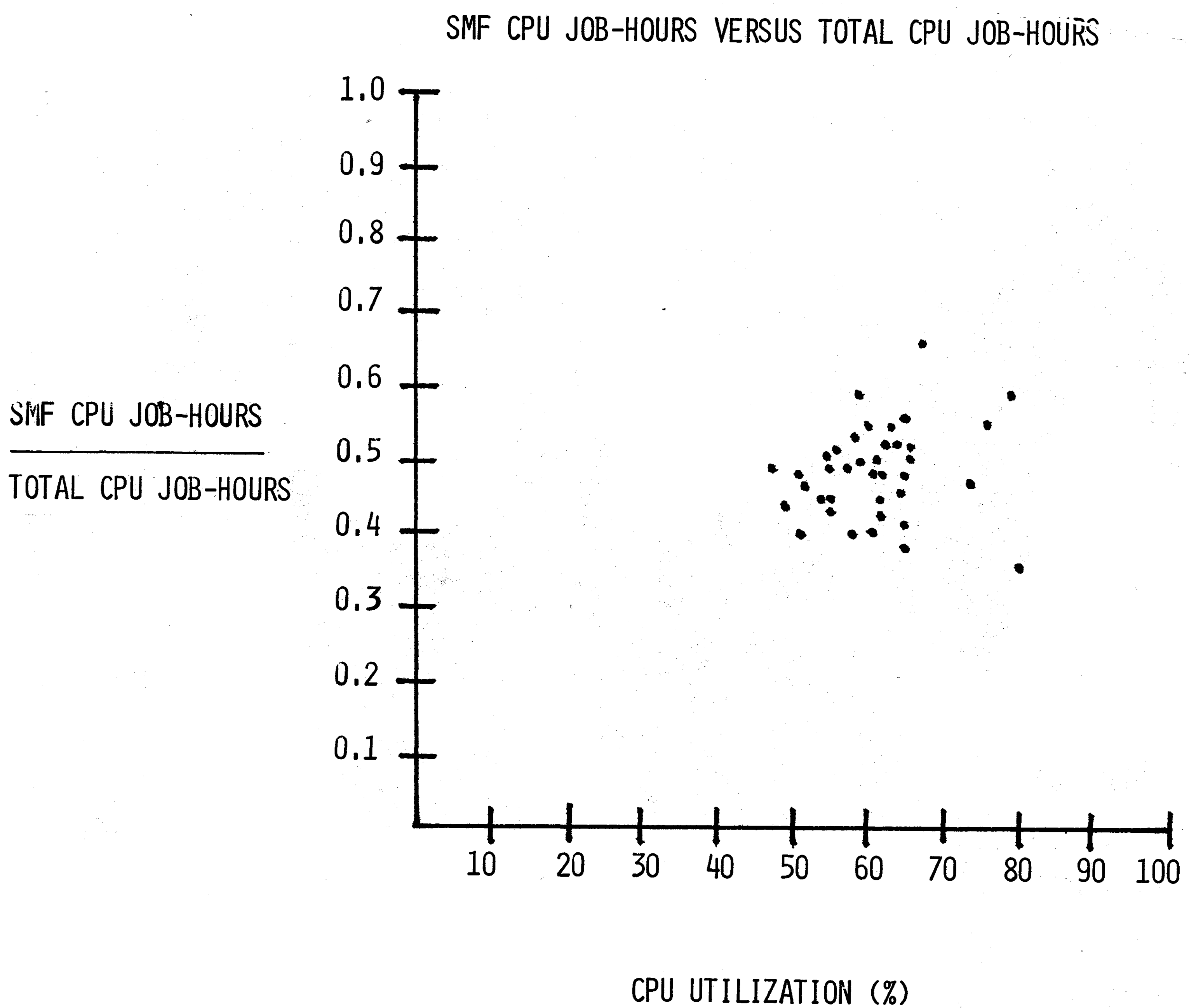
O AFTER A CERTAIN LEVEL OF CONFIDENCE IS ESTABLISHED IN MEASURED PARAMETERS

- o LOADING
- o CPU SERVICE (APPLICATION JOB-HOURS)
- o RESOURCE UTILIZATION
- o RESPONSE/TURNAROUND TIMES

ONE MAY BEGIN TO MOVE TO SIMPLE QUEUEING MODELS WHICH WILL HOPEFULLY ENHANCE PROJECTIONS



CPU CONSUMPTION BY APPLICATION


SMF CPU JOB-HOURS VERSUS TOTAL CPU JOB-HOURS
SMF CPU JOB-HOURS
TOTAL CPU JOB-HOURS
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
10
20
30
40
50
60
70
80
90
100
CPU UTILIZATION (%)

FORECASTING

WORKLOAD (BY APPLICATION GROUP)

- o JOBS/HOUR
- o TRANSACTIONS/MINUTE

SERVICE BY WORKUNIT
(FOR CRITICAL JOBS/APPLICATIONS)

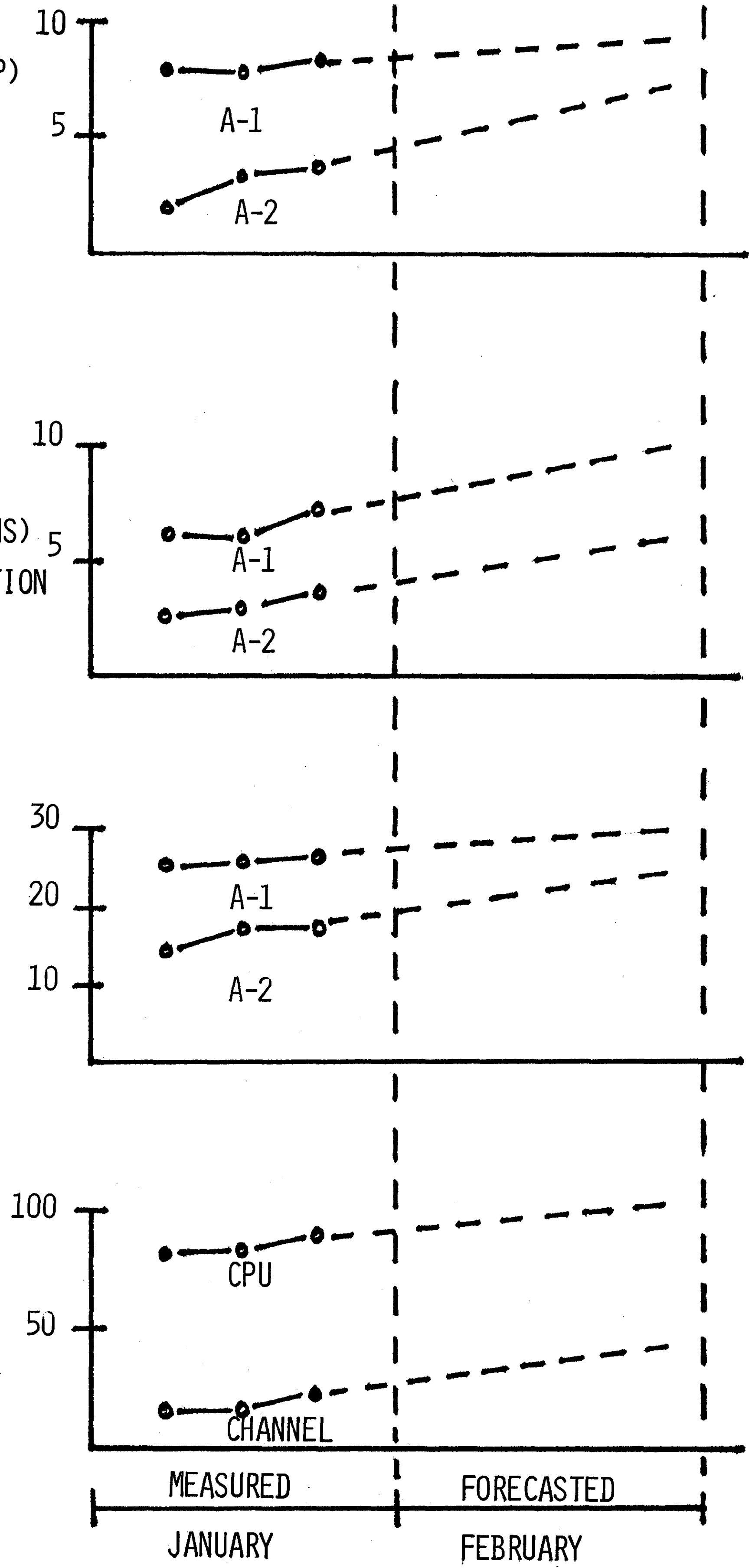
- o RESPONSE TIME BY TRANSACTION
- o TURNAROUND TIME BY JOB

TOTAL CPU SERVICE
(BY APPLICATION AND SUMMARIZED
BY TIME PERIOD-DAY, WEEK,
MONTH)

- o CPU JOB-HOURS

UTILIZATION (%)

- o CPU
- o CHANNEL
- o CRITICAL I/O DEVICES



PREDICTION/FORECASTING METHOD

- 0 A DETAILED UNDERSTANDING OF THE SYSTEM IS ESSENTIAL, OBTAINED VIA THE CAPACITY PLANNING METHOD.

- 0 EMPIRICAL ANALYSIS - MODELS/EQUATIONS AND INSIGHTS OBTAINED FROM CLOSE OBSERVATION AND MEASUREMENT OF SYSTEM PARAMETERS.
  - o HARDWARE SATURATING PROPERTIES
  - o SOFTWARE SATURATING PROPERTIES
  - o MAIN MEMORY REQUIREMENTS

- 0 DATA ANALYSIS
  - o VALIDATION
  - o TRENDS
  - o CORRELATION
    - o LOADING
    - o UTILIZATION
    - o RESPONSE/TURNAROUND

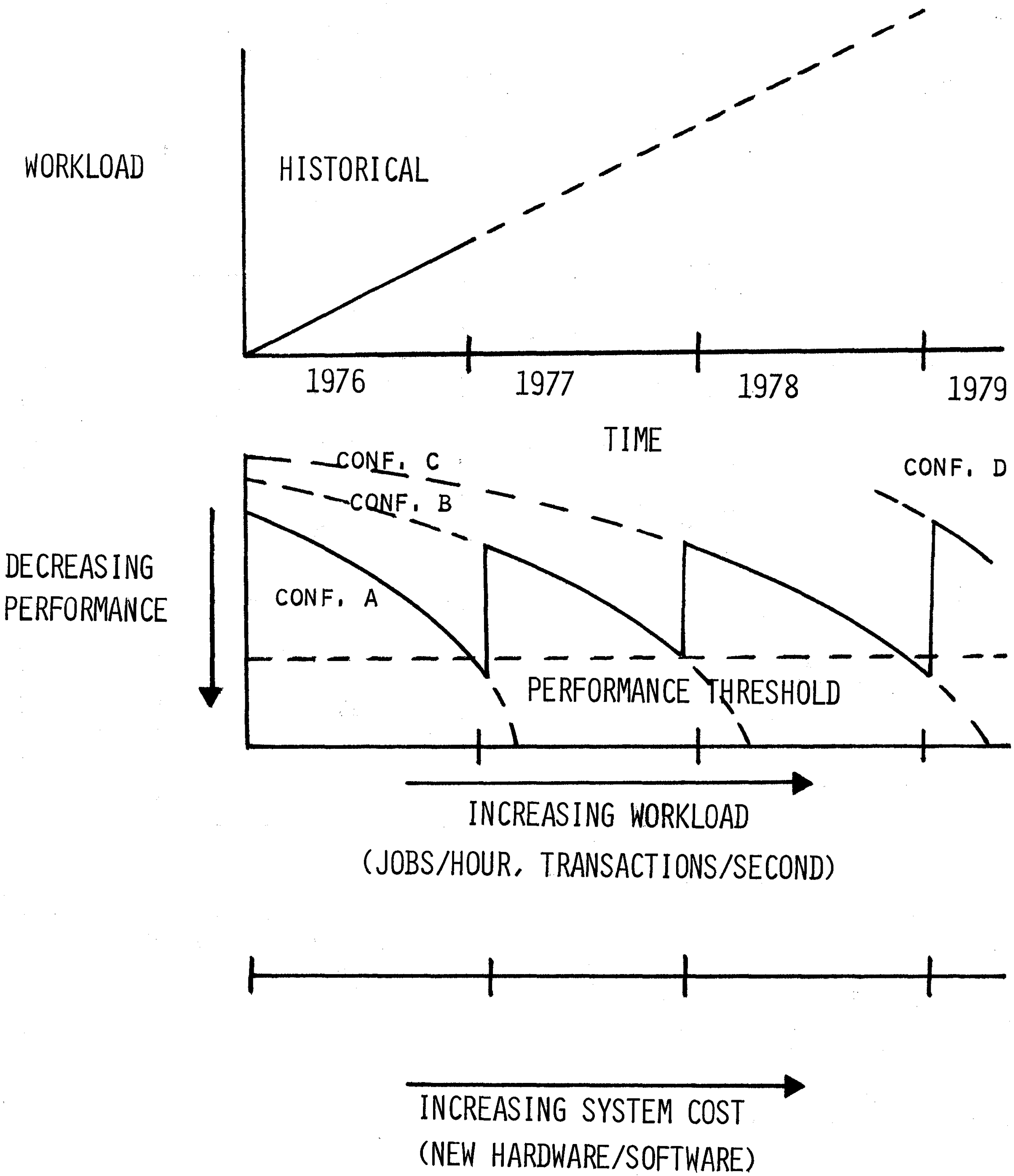
- 0 HAND CALCULATIONS/RULES OF THUMB

- 0 SINGLE SERVER QUEUEING MODELS

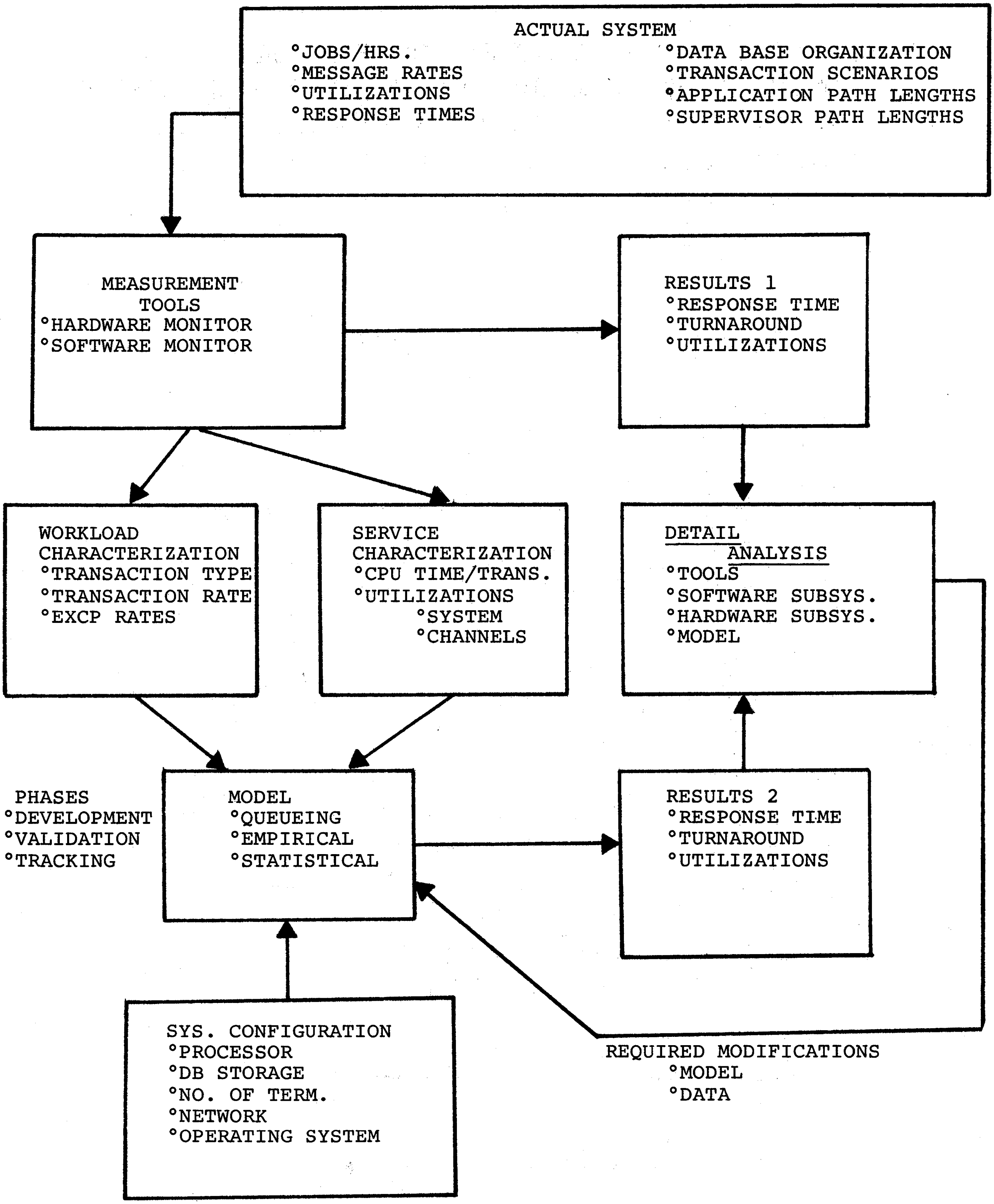
- 0 VALIDATION/CONFIDENCE FACTOR
  - o ACTUAL OPERATION
  - o BENCHMARKS

PREDICTION/FORECASTING PROCESS

HAVING A PROJECTED LOAD AND AN UNDERSTANDING OF THE LOADING SCENARIO (CPU, CHANNELS, CONTROL UNITS, ETC.), VIA THE CAPACITY PLANNING METHOD, NEW EQUIPMENT REQUIREMENT DATES CAN BE GROSSLY PREDICTED. THESE PREDICTIONS WILL BE BASED ON CUSTOMER DESIRED PERFORMANCE LEVELS.



PERFORMANCE PREDICTION CYCLE

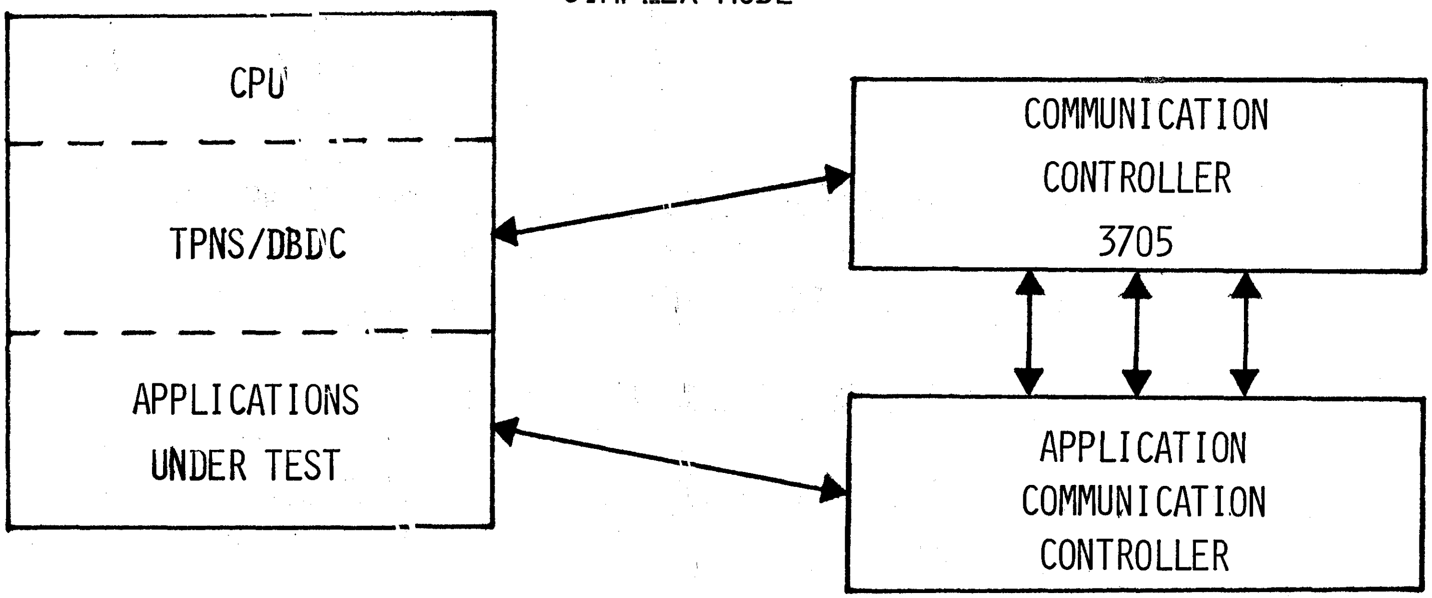


PREDICTION/FORECASTING METHOD

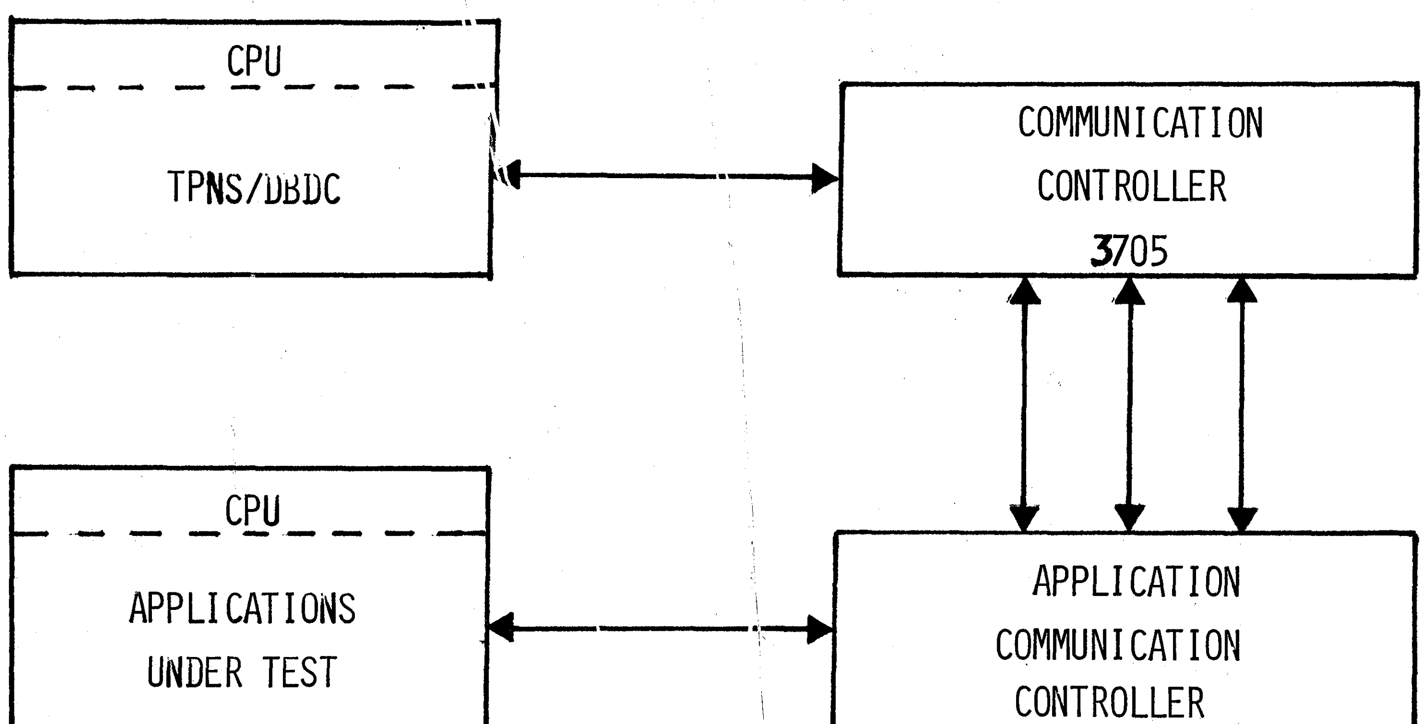
- O DRIVERS
  - o SIMULATES/REPLACE A USER DEFINED TP NETWORK
  - o TRANSPARENT TO USER APPLICATION
  - o INCLUDES A "TERMINAL SCRIPT"
  - o SIMULATE CURRENT SYSTEM IN A CONTROLLED ENVIRONMENT
  - o SIMULATE FUTURE ENVIRONMENTS
  - o PRODUCE OUTPUT FOR MODELING SOME PROPOSED CONFIGURATION (TRANSACTION LOADING AND SERVICE DATA)
- O IBM DRIVERS
  - o TPNS
  - o DBDC

USE OF DRIVERS FOR PERFORMANCE PREDICTION

SIMPLEX MODE



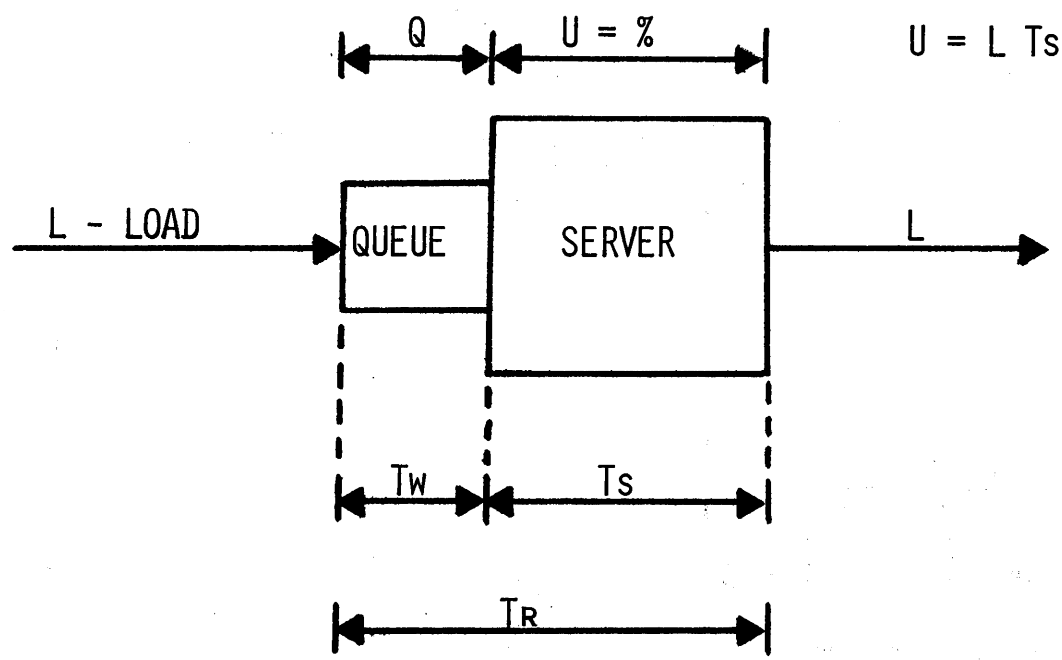
DUPLEX MODE



QUEUEING ANALYSIS

SINGLE SERVER QUEUEING MODEL

RESOURCES: CPU, CHANNEL, I/O DEVICE, ETC.

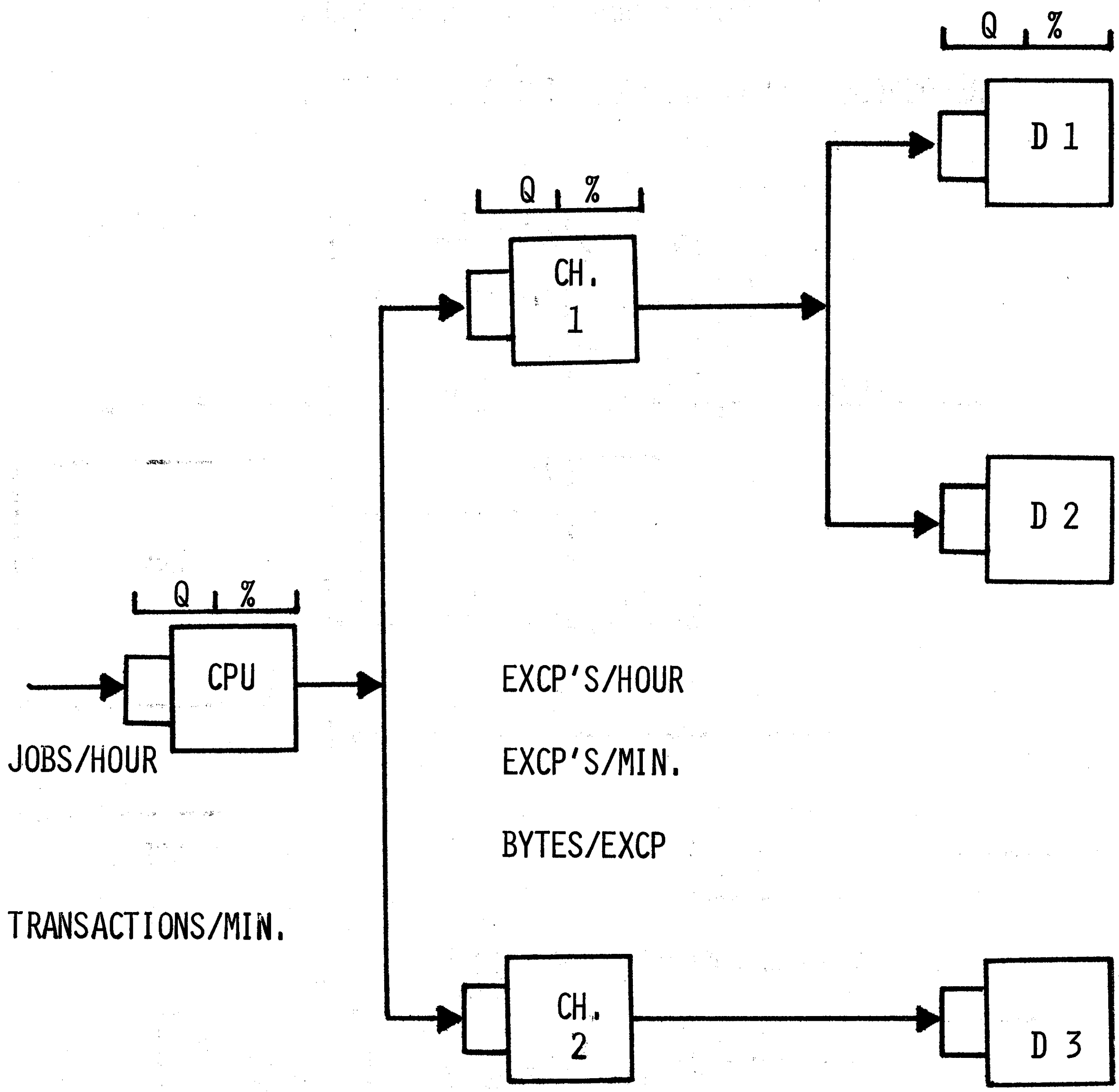


M/M/1: $T_W = \frac{T_S\,U}{1-U}$

M/G/1: $T_W = \frac{U\,T_S}{2\,(1-U)}\left[1 + \frac{\sigma_S^2}{T_S^2}\right]$

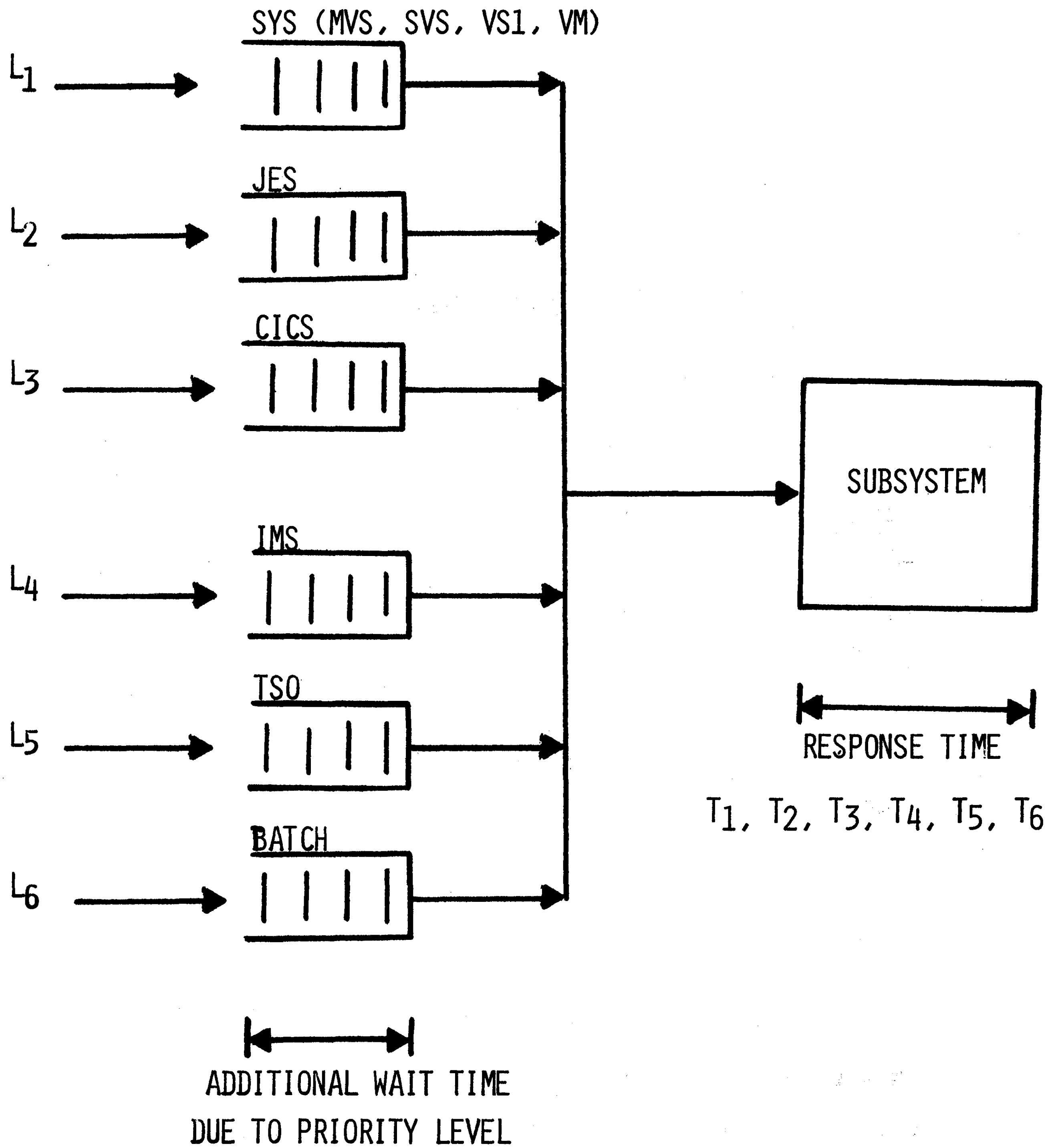
$T_R = T_W + T_S$

LOADING, SERVICE, ELAPSED



NOTE: CONTROL UNIT CHARACTERISTICS ARE ALSO STUDIED AS PART OF THIS CONFIGURATION.

COMBINE SUBSYSTEMS FOR TOTAL SYSTEM ANALYSIS



CONCLUSIONS

CONCLUSIONS

- O CAPACITY PLANNING REQUIRES A MAJOR COMMITMENT ON THE PART OF THE DP INSTALLATION

- O METHODOLOGY APPEARS SOUND

  - o DEVELOP AN ORGANIZING STRUCTURE
  - o INITIATE EFFORT WITH A SMALL NUMBER OF MEASUREMENT TOOLS (2-4)
  - o TRACK AND PLOT PERFORMANCE ON A CONTINUING BASIS
  - o DRIVING FORCE IS USER SERVICE OBJECTIVES
  - o MOVE TO COMPLEX MODELLING TECHNIQUES ON A TIMELY BASIS

- O BENEFITS OF CAPACITY PLANNING PROCESS

  - o IMMEDIATE
  - o LONG RANGE

- O A VIABLE CAPACITY PLANNING PROGRAM IS PARAMOUNT FOR UNDERSTANDING AND MANAGING TODAY'S COMPLEX DATA PROCESSING ENVIRONMENT

READER'S COMMENTS

Title: An Introduction to Capacity Planning
Washington Systems Center
Technical Bulletin GG22-9001-00

Please state your occupation: ______________________________

Comments:

Please mail to: LeeRoy Bronner
IBM Corporation
18100 Frederick Pike
Gaithersburg, Md. 20760