FIPS PUB 75

FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION

1980 SEPTEMBER 18

U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards

GUIDELINE ON CONSTRUCTING BENCHMARKS FOR ADP SYSTEM ACQUISITIONS

CATEGORY: ADP OPERATIONS

SUBCATEGORY: BENCHMARKING FOR COMPUTER SELECTION

U.S. DEPARTMENT OF COMMERCE, Philip M. Klutznick, *Secretary*
Jordan J. Baruch, *Assistant Secretary for Productivity, Technology and Innovation*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

Foreword

The Federal Information Processing Standards Publication Series of the National Bureau of Standards (NBS) officially publishes Federal standards and guidelines adopted and promulgated under the provisions of Public Law 89-306 (Brooks Act) and under Part 6 of Title 15, Code of Federal Regulations. Under P.L. 89-306, the Secretary of Commerce has important responsibilities for improving the utilization and effectiveness of computer systems in the Federal Government. In order to carry out the Secretary's responsibilities, the NBS, through its Institute for Computer Sciences and Technology, provides leadership, technical guidance, and coordination of Government efforts in the development of technical guidelines and standards in these areas.

The successful outcome of most ADP system acquisition efforts is largely determined by the effective identification and representation of an agency's requirements through benchmarks. Benchmark construction, however, has proved to be a very costly process within the Federal Government, in part, because no general guidance exists on how to do it. It is hoped that this Guideline will help procuring agencies reduce the cost of constructing benchmarks, as well as improve their representativeness so that the risks in acquiring computer systems are reduced. To this end, the National Bureau of Standards is pleased to make this Guideline on benchmark construction available for use by Federal agencies in the ADP system acquisition process.

James H. Burrows, *Director*
Institute for Computer Sciences
and Technology

Abstract

This Guideline describes a step-by-step procedure for constructing benchmarks for use in the acquisition of ADP systems. Ten steps in the benchmark construction process are identified involving such areas as workload analysis and forecasting, construction of the benchmark mix, and documentation and testing of the benchmark package. Although the Guideline is directed to the technical staff who will actually be constructing the benchmark, portions of it should also be useful to management. In addition, the Guideline should be useful to those in private industry who are also involved in constructing benchmarks for use in the evaluation of alternative vendor systems.

Key words: ADP acquisition; ADP procurement; benchmarking; Federal Information Processing Standards Publication; performance evaluation; workload analysis; workload characterization; workload representation.

Nat. Bur. Stand. (U.S.), Fed. Info. Process. Stand. Publ. (FIPS PUB) 75, 43 pages.
(1980)
CODEN:FIPPAT

For sale by the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

Federal Information
Processing Standards Publication 75

1980 September 18

ANNOUNCING THE

GUIDELINE ON CONSTRUCTING BENCHMARKS FOR ADP SYSTEM ACQUISITIONS

Federal Information Processing Standards Publications are issued by the National Bureau of Standards pursuant to the Federal Property and Administrative Services Act of 1949, as amended, Public Law 89-306 (79 Stat. 1127), as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973), and Part 6 of Title 15 CFR (Code of Federal Regulations).

Name of Guideline: Guideline on Constructing Benchmarks for ADP System Acquisitions.

Category of Guideline: ADP Operations.

Subcategory of Guideline: Benchmarking for Computer Selection.

Explanation: This Guideline describes a step-by-step procedure for constructing benchmarks during the competitive acquisition of ADP systems. It identifies the "best practices" found within private industry and the Federal Government with respect to benchmark construction.

Approving Authority: U.S. Department of Commerce, National Bureau of Standards (Institute for Computer Sciences and Technology).

Maintenance Agency: U.S. Department of Commerce, National Bureau of Standards (Institute for Computer Sciences and Technology).

Cross Index: Federal Information Processing Standards Publication (FIPS PUB) 42-1, Guidelines for Benchmarking ADP Systems in the Competitive Procurement Environment.

Applicability: This document is intended as a basic reference guide for constructing benchmarks during the competitive acquisition of computer systems. Its use is generally applicable throughout the Federal Government, as well as in private industry.

Qualifications: This Guideline represents "best practices" for benchmark construction based upon input received from sources both within and outside of the Federal Government.

The purpose of this Guideline is to recommend an orderly process for benchmark construction in order to help reduce the costs and the risks in agency acquisition efforts. The guidance herein is intended for use during the competitive acquisition of ADP systems and does not address the problems of benchmark construction during the acquisition of ADP services, as, for example, through the Teleprocessing Services Program (TSP). It does not attempt to address every contingency of benchmark construction; thus, specific decisions and actions will vary from agency to agency. Furthermore, this Guideline does not address other uses of benchmarking, such as in capacity planning, nor does it address other parts of the ADP acquisition process, such as contractual safeguards, procurement regulations and policy, Federal ADP management policy, validation of Federal standards or other ADP procurement considerations. Thus, in order to be consistent with overall Federal policy, the user should seek current guidance from applicable Office of Management and Budget (OMB) and General Services Administration (GSA) policy and procurement directives.

The extent to which this Guideline should be followed is dependent upon such factors as the expected dollar value of the acquisition, available workload data, staff, and system resources, and the criticality of the agency mission(s) to be supported by the new or replacement system.

This document will need to be expanded or otherwise modified as further research is conducted and knowledge obtained on benchmark construction. Comments, critiques, and technical contributions directed to this end are invited. These should be addressed to the Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, Washington, DC 20234.

Where to Obtain Copies of the Guideline: Copies of this publication are for sale by the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. When ordering, refer to Federal Information Processing Standards Publication 75 (FIPS-PUB-75), and title. When microfiche is desired, this should be specified. Payment may be made by check, money order, purchase order, or deposit account.

Federal Information
Processing Standards Publication 75

1980 September 18

Specifications for

GUIDELINE ON CONSTRUCTING BENCHMARKS FOR ADP SYSTEM ACQUISITIONS

CONTENTS

INTRODUCTION

A. Purpose

Benchmarking is an accepted method for testing vendor systems during the competitive acquisition of computer systems within both private industry and the Federal Government. The success of benchmarking as an evaluation technique, however, depends upon the extent to which benchmarks can be constructed that are representative of expected, future workloads.

The immediate purpose of this Guideline is to describe a step-by-step procedure for the construction of benchmarks for use during the competitive acquisition of ADP systems which support batch and/or online workloads. This Guideline identifies the "best practices" found within private industry and the Federal Government with respect to benchmark construction. Detailed guidance is given when possible and where appropriate. In some areas, few established practices exist, thus treatment of these areas will be more general. The ultimate purpose of this Guideline is to help reduce the cost of the often lengthy benchmark construction process in the Federal Government, while also helping agencies construct representative benchmarks, and, indirectly, helping to reduce the vendors' time and cost of implementing the benchmark.

Interviews conducted by the National Bureau of Standards (NBS) with Federal agencies and private industry indicate a definite need for a procedural guideline on benchmark construction. It is hoped that this Guideline will enable procuring agencies to construct representative benchmarks, to the maximum possible extent, in order to minimize the risks in their evaluation of vendor-proposed ADP systems.

No general guideline of this kind can address every contingency; thus, specific decisions and actions in support of the benchmark construction will vary from agency to agency. Furthermore, the extent to which this Guideline should be followed depends upon such factors as the expected dollar value of the acquisition, the availability and reliability of workload information, the availability of staff and system resources for constructing the benchmark, and the criticality of the agency mission(s) to be supported by the new system. The reader is referred to applicable Federal Procurement Regulations (FPR's) concerning the use and appropriateness of benchmarks for various dollar-value procurements.

This Guideline is not a replacement of FIPS PUB 42-1 ("Guidelines for Benchmarking ADP Systems in the Competitive Procurement Environment" [NBS 77]). Rather, it provides procedural guidance for constructing the benchmark that would be used for evaluating vendor systems under the guidelines set forth in FIPS PUB 42-1. This Guideline does not address the problems of constructing benchmarks during the acquisition of ADP services, as, for example, through the Teleprocessing Services Program (TSP); nor does it address the use of benchmarks during acceptance testing or at the time system augmentations are due to occur. Future guidelines are expected to address these areas. Also, this Guideline does not address the maintenance of benchmarks—that is, maintaining the currency of benchmarks over periods of time (e.g., during long delays in the acquisition process when future requirements may change). Furthermore, although this Guideline addresses benchmark construction techniques for both batch and online workloads, the reader is referred to "Use and Specifications of Remote Terminal Emulation in ADP System Acquisitions" [GSA 79] for further information on when and how to use remote terminal emulation during the acquisition of systems requiring an online component.

A "benchmark" is defined in this Guideline as one or more "benchmark mixes," together with rules for running each mix during a Live Test Demonstration (LTD) on vendor-proposed systems. A "benchmark mix" is defined to be a collection of "benchmark problems" (i.e., batch programs and online activities) together with the terminal designations for online activities, the sequence of benchmark problems, data requirements, etc. that, ideally, are representative of some future workload requirements. The "benchmark (or LTD) rules" define the operational requirements associated with running the benchmark mix. A "timed benchmark test" is used to test the capability of a computer system to perform within certain predetermined service level requirements. Ideally, a timed benchmark test should produce the same performance characteristics on the system under test as would the real workload it represents. Although a benchmark may also be used during a "functional demonstration" to verify that a system has certain functional capabilities, this Guideline will focus primarily on procedures for constructing timed benchmark tests.

Portions of this Guideline are directed to technical staff, operations management, and top management. The Introduction and Overview sections, as well as portions of STEP 1, will be useful to top management. Technical staff and operations management, who will actually be responsible for constructing the benchmark, should find the entire document useful.

B. Common Uses of Benchmarking

Although benchmarking is generally thought of as an important and necessary tool during the acquisition process, it also has many other useful applications:

1. The effects of software and hardware changes on system performance can be evaluated by running a representative benchmark before and after such changes.
2. Benchmarking can be used in capacity planning to determine the unused capacity and the saturation point of the present system. This is done by first constructing a benchmark to represent projected workload(s) and then running the benchmark to stress test the current system; i.e., to determine at what load levels required service levels can no longer be attained. This application of benchmarking would thus enable an agency to plan better for future acquisitions.
3. Benchmarking can also be used to evaluate the design of computer systems. This application is largely used by the vendors themselves. Computer system designers often use benchmarks to evaluate the capabilities and performance of their new systems.
4. Benchmarking is most commonly used as an evaluation technique in the ADP acquisition process. It is a common test by which different vendor systems can be evaluated. Benchmarking in this context can serve several important functions. It can assist the vendors in determining the most cost effective offering to satisfy the agency's requirements. It can facilitate the verification of the proposed system as to the time required to perform the workload and as to its functional capabilities. And, finally, it can sometimes be used prior to or during acceptance testing, after award, to verify that the delivered system is consistent with the system benchmarked during the evaluation phase.

This fourth application of benchmarking is the subject of this Guideline.

C. Background

A detailed description of the competitive ADP system acquisition process is not within the scope of this Guideline; however, it is important to identify how the benchmark construction process fits into the total acquisition process within the Federal Government. In general, the ADP system acquisition process involves six main components:

1. **Studies and Approvals.** Feasibility studies, approvals, sharing and consolidation studies, funding studies, etc. are generally performed as the first step in the acquisition process, often in response to internal and/or external regulations.
2. **Definition of User Requirements and Technical Specifications.** User requirements provide the basis for the Request for Proposals (RFP), and for the

evaluation and selection procedures. Development of technical specifications (based on user requirements), which will be released to all interested vendors, is a crucial part of the process. The specifications should be general enough to assure wide competition, yet specific enough to delineate user requirements. Mandatory requirements are those user requirements that the procuring agency identifies as necessary for the completion of its mission. Desirable features are those agency-specified options that have value to the agency, but are not essential to the completion of its mission, or that could also be obtained through some other source (as in the case of some application software packages).

3. **Evaluation Plan and Strategy.** An evaluation plan is devised that defines the cost and technical factors that are to be evaluated and the strategy for conducting the evaluation (how they are to be evaluated, alternative means of evaluation, relative importance of each factor). With regard to the technical evaluation, a questionnaire is usually devised as a common format for determining the extent to which each vendor's offering meets the technical specifications in the RFP. As part of the evaluation plan, the objectives of the benchmark should be clearly defined—that is, the agency requirements or technical specifications that the benchmark is intended to test (which cannot be tested through other means) and the method for testing them (either through a timed benchmark test or by means of a functional demonstration). Once the benchmark objectives are defined, benchmark construction takes place and the benchmark package is developed.

4. **Preparation and Release of the RFP.** The RFP combines the user requirements and technical specifications with the evaluation criteria, benchmark package, and contractual requirements. The RFP is released, usually soon followed by vendor questions and subsequent amendments to the RFP.

5. **Evaluation of Proposals.** Proposal evaluation is the process by which the procuring agency determines the extent to which the hardware and software configurations proposed by the vendors meet the mandatory requirements and the desirable features stated in the RFP. Benchmarking is a step in this evaluation process designed to validate the vendor's response to those mandatory requirements and desirable features (if proposed by the vendor) that cannot be sufficiently evaluated from the vendor's written proposal. The most common form of benchmarking is the testing of the vendor's proposed system capabilities in terms of minimum, specified service requirements, such as turnaround time and response time.

6. **Selection and Contract Award.** After an evaluation of each vendor's proposal and (where appropriate) performance during benchmark testing, negotiations are held with qualifying vendors at the end of which best and final offers are usually solicited. A contract is then awarded to the vendor who meets the mandatory requirements in the RFP, and offers a system that is most advantageous to the procuring agency in terms of technical capabilities and expected life cycle costs.

OVERVIEW

This section defines several important terms and concepts that will be used throughout the rest of this Guideline. A glossary at the end of the Guideline provides summary definitions for these and other terms. This section also provides a brief outline of the benchmark construction process, which will be discussed in more detail in succeeding sections.

A. Terminology

1. General

The term **application** will be used here to mean a logically distinct, identifiable problem presented to a computer system. Hence, "application" will be used to refer to what is typically called a "job-step" or "online activity." The term **application system** will denote a collection of related applications, the purpose of which is to perform a distinct agency function (e.g., payroll). The term **workload** will refer to a collection of agency applications. For online workloads, the term **online session** will denote all **online activities** performed between logon and logoff, where an online activity consists of a series of logically related **online commands**.

2. ADP Requirements

The complete description of an application will be termed the application's **ADP requirements**. In order to emphasize the special meaning of this term, each instance of its use will be italicized. Table 1 contains a list of typical *ADP requirements*, which are briefly discussed below. The *ADP requirements* in Table 1 will be used later to characterize present and future workloads. The term **workload requirements** will refer to the collection of *ADP requirements* for all applications of a given workload, and the term **support requirements** will mean such global requirements as communications equipment, accounting logs, performance monitors, backup and recovery capabilities, etc.

Table 1. *ADP Requirements*

- Processing Demands
  - ADP operations
  - resource usage
- Processing Mode/Type
- Shift executed
- Priority
- Application dependencies
- Security requirements
- Vendor-supplied vs. user-written
- Data files characteristics (number, source, structure, and size)
- Input/output volumes (e.g., number and types of transactions processed, input records, etc.)
- For batch applications:
  - service requirements (e.g., turnaround time)
- For online applications:
  - service requirements (e.g., response time)
  - terminal speed and type
  - think times and typing times
  - number of concurrent users

334-807 O - 81 - 2

a. Processing Demands

An application can be described in terms of the organizational functions it supports, the ADP operations it performs, or the system resources it consumes.

At the highest level, an application can be described in terms of the organizational functions it is intended to support. This type of description does not depend on the incumbent computer system; hence, it can be referred to as a system-independent description. Examples of organizational functions are payroll, inventory, finance and accounting, and engineering. The major disadvantage of describing applications only in this manner is that this description is at too high a level for constructing benchmark problems.

An application can also be described by the ADP operations it performs. Examples of ADP operations are:

COBOL compile,
FORTRAN compile,
sort,
report generation,
database update,
database query,
online commands.

Furthermore, a number of parameters can be associated with each ADP operation. For example, typical parameters for the ADP operation "sort" might include:

number of records sorted,
size of records sorted,
number of sort keys,
size of sort keys.

The description of an application by ADP operations is also considered to be system-independent and would represent an ideal description of the workload. However, it is often difficult to obtain, in an automated way, the parameter data associated with a given ADP operation.

On the lowest level, an application can be described by the resources it consumes. A computer system can be considered as a collection of resources upon which an application places demands. For example, consider the following resources common to many computer installations:

CPU,
I/O channels and devices,
memory,
unit record devices,
communications equipment.

The demands placed on these resources by an application can be quantified by such parameters as:

CPU time,
I/O usage (number of I/O activities, number and type of devices used, channel time),
memory allocated and/or used,
unit device activity and volume,
number of characters transmitted/received.

The demands on these resources can be considered, collectively, as a characteristic of the application processed by a specific computer system. Because this description will vary from one computer system to another, this level of description is considered to be system-dependent. (In some cases where an application's memory requirement for data far exceeds the system-dependent memory size for its machine language instructions, then memory size can sometimes be used as a system-independent measure.) The advantage of describing an application by its resource demands is that this is the level at which the most data exists. However, it is also the level that is most system dependent and, hence, least desirable.

Because the demands that an application places on a computer system can be described in terms of either ADP operations or resource usage, these two types of descriptions will be termed an application's **processing demands**, and are considered part of an application's overall *ADP requirements* (see Table 1).

b. Other *ADP Requirements*

Each application can also be described by other characteristics: its **processing mode/type**—that is, the manner in which it is presented to and executes on a computer system (Table 2 contains a list of common processing modes and types); the shift during which the application is executed; its priority; its dependency on other applications; its service requirements; etc.

Table 2

Processing mode	Processing type
Batch:	Initiated at the central site
	Remote batch
	Online initiated
Online:	Interactive program development
	Interactive program execution
	Text processing
	Database query/update
	Data entry
	Interactive graphics
	Transaction-oriented

B. Outline of the Benchmark Construction Process

This Guideline describes a step-by-step approach to benchmark construction. The following ten procedural steps have been identified:

STEP 1. Define Benchmarking Objectives and Complete Preliminary Activities. This step discusses the selection of the benchmark team and the importance of having certain preliminary activities completed, as well as having definite objectives and goals defined prior to the benchmark construction.

STEP 2. Quantify the Present Workload Requirements. This step identifies commonly available sources of data for quantifying the *ADP requirements* of present applications (i.e., for taking an "inventory" of present applications); it introduces the concept of "application groups" as a way of grouping applications; and, finally, it discusses the association of application groups with distinct, organizational entities.

STEP 3. Survey Users. This step discusses user surveys as a source for obtaining additional information on present applications, as well as for obtaining user forecasts of new or changing application systems.

STEP 4. Forecast Future Workload Requirements. This step describes methods for quantifying aggregate future workload *ADP requirements*, for determining bounds on the workload forecasts, and for identifying potential augmentation points.

STEP 5. Categorize Future Workloads. This step presents a technique for partitioning the workload into distinct categories for each potential augmentation point.

STEP 6. Determine Relative Contribution of Each Category. This step discusses methods for determining the relative contribution of each category obtained in STEP 5.

STEP 7. Scale Each Category. This step presents a technique for determining the running times (using the results from STEP 6) for each benchmark problem that will represent the categories identified in STEP 5.

STEP 8. Represent Workload Categories with Benchmark Problems. This step discusses the selection of real or synthetic programs to represent the batch workload categories from STEP 5, as well as methods for representing the online categories and constructing the benchmark mix(es).

STEP 9. Fine Tune Each Benchmark Mix on the Present System. This step discusses the advantages, as well as some possible problems, in testing the benchmark mix(es) on the present system.

STEP 10. Prepare the Benchmark Package and Test the Benchmark. This step discusses the benchmark package (i.e., the documentation of each benchmark mix and the LTD rules), as well as ways of testing the benchmark by running the benchmark mix(es) on other systems.

The remainder of this Guideline discusses in detail each of the above benchmark construction steps.

BENCHMARK CONSTRUCTION STEPS

STEP 1. Define Benchmarking Objectives and Complete Preliminary Activities

Once it has been decided that benchmarking will be used, its objectives and the role it is to play in the evaluation process should be clearly defined. What agency requirements or technical specifications is the benchmark designed to test? Which of these are to be tested by a timed benchmark test and which will be tested by a functional demonstration? Are there other ways of testing these capabilities? Should they be used in place of or as a consistency check on the benchmark? All of these questions should be addressed in the context of the total evaluation effort.

A great deal of information about the nature of *ADP requirements* should be collected and refined continuously throughout the ADP system life cycle. Examples are user service requirements, functional requirements, growth of existing workload, etc. For this reason, the following activities should, ideally, have been completed prior to the benchmark construction:

1. definition of the agency's service requirements;
2. determination of the new system's life;
3. definition of various operational requirements;
4. determination of a system design concept;
5. forecast of future workload requirements.

Unfortunately, many of these activities are often not completed (or even begun) until a benchmark is about to be constructed. The specific tasks to be performed during the benchmark construction effort will thus depend on the extent to which these actions and management decisions have already been made. The agency should therefore determine which of these activities will be accomplished by the benchmark team and which have been or will be accomplished by other organizational entities.

This step discusses the first four of the above activities, as well as the nature of the benchmark team itself; STEP's 2 through 4 discuss the development of future workload requirements.

1.1 Establish the Benchmark Team

The benchmark team will vary in size and composition from agency to agency and from one procurement to another. There are several factors that determine the exact size and composition of the benchmark team. These factors include the size of the procurement, the system concept to be employed, and the variety of existing and new agency functions. Usually, the benchmark team is composed of hardware/software specialists, users, and other technical personnel familiar with user applications. Programmers may be used to help develop and test the benchmark problems. Database specialists (including, perhaps, even the Database Administrator) may be used to help determine the agency's database requirements. A statistician could be used during the workload quantification, user survey, workload forecast, and workload categorization steps (STEP's 2 through 5) to assist in the use of various data reduction and analysis techniques. A telecommunications specialist should be added if the new system is expected to include teleprocessing. The benchmark construction team leader should be a hardware/software specialist thoroughly familiar with the present system's capabilities, with the organization's future requirements, and with state-of-the-art system capabilities.

As indicated above, a number of activities and management decisions should be made before the actual benchmark construction can be initiated. These decisions serve as design criteria for the personnel actually doing the benchmark construction. Good cooperation between functional and technical personnel is important in the decision-making process to assure that user requirements are appropriately identified. Several of these preliminary decisions are discussed below.

1.2 Define Service Orientation

The ADP service orientation taken by an agency in fulfilling its mission can be production-oriented, user-oriented, or a combination of these. In a production environment, the agency attempts to use the capabilities of the new system to the maximum extent possible in terms of throughput. In a user-oriented environment, service to the users has higher priority over throughput. That is, responsiveness to each job's demands is more important than the total number of jobs processed per unit of time. Both objectives may thus be expressed in terms of time, either time to process a certain number of jobs (production-oriented), or time to respond to individual user requests (user-oriented). Often, one objective (either production- or user-oriented) is dominant during a given time-frame; e.g., from 8 a.m. to 4 p.m. user-dominated processing may exist, and from 4 p.m. to midnight production-dominated processing may occur. The orientation taken by an agency will affect the specific kinds of service requirements reflected in the benchmark.

As seen earlier, an application can be described by its mode of processing, either batch or online. Service requirements will most likely not be the same for all applications, even for ones of the same processing type. For example, there are two types of batch workloads initiated at a central site: scheduled and unscheduled. The service requirements of a scheduled batch workload are based on meeting the schedule (e.g., daily, weekly, or monthly reports) and can be defined in terms of pre-defined deadlines. The service requirements of an unscheduled batch workload, which would typically involve work such as program development, can be defined in terms of turnaround time goals. For online applications, service requirements are usually defined by response time.

The kinds of service requirements selected must be in accordance with the objectives defined: production, user-oriented, or a combination of the two. For combination objectives, the priority between objectives should be identified, especially if the production- and user-oriented service objectives occur within the same time-frame. This is accomplished as part of a survey of agency functions (STEP 3) in which applications can be categorized as production-oriented or user-oriented, and in which the criticality of each application to the agency's mission is determined. The service requirements associated with the critical applications are then weighed against the (possibly) competing service requirements associated with the noncritical applications to determine the type of service to be met by the new system and reflected in the benchmark.

1.3 Determine System Life

As a design criterion to benchmark construction, the expected life of the new system must also be defined. This is the same period of time for which future requirements will later be analyzed and projected.

1.4 Define Operational Requirements

Future operational requirements, such as the number of shifts per week expected on the new system, must be determined because any changes in the number or size of shifts would affect the number of operational hours available on the new system—a parameter needed in STEP 7 where the workload categories are scaled.

1.5 Determine System Design Concept

Prior to the actual benchmark construction, it may be necessary to evaluate several alternative system design concepts (e.g., centralization versus decentralization). The reader is referred to OMB Circular A-109 [OMB 76] for policies to be followed by agencies in the acquisition of major systems, including the evaluation of alternative system design concepts. In many cases, the analysis of alternative system design concepts may require more information on the characteristics of the future workload, which will only be known after STEP 4, below.

STEP 2. Quantify the Present Workload Requirements

The quantification of present workload requirements is one of the first steps in the benchmark construction process since forecasts of future requirements will be based on information about the present workload. Benchmark mixes will then be constructed to represent selected periods of the total workload forecast. When no present workload exists and the system being acquired will be used for new applications, this step will initially be bypassed. In this case, STEP 3 will first be performed to obtain information from the users about their future requirements. If existing applications (even if on another system) can be identified similar to these new applications, then STEP 2 might be entered to obtain additional detail about these similar applications, and hence about the new applications' expected *ADP requirements*.

STEP 2 generally involves quantifying two aspects of the present workload for some recent time-frame: support requirements and application *ADP requirements*. The time-frame to be selected should be long enough to include all present applications and any cyclical changes in their characteristics. Also, if the agency's mission requires the processing of peak activities within specified service requirements, then the time-frame may include these peak periods. Support requirements to be quantified during the selected time-frame include such items as aggregate data storage needs, unit record equipment, and terminal and network characteristics. Information on such global workload requirements can usually be obtained from system catalogs, accounting log files, user surveys, and communication monitors.

Quantifying the *ADP requirements* of applications running on the current system during the selected time-frame usually requires more than one source of data, and more than one analysis technique. The remaining portion of this section discusses how such a quantification can be performed. Care should be taken that certain events which took place during the time-frame being analyzed do not invalidate the data being collected. For example, events such as job reruns can give a false picture of previous processing characteristics.

2.1 Inventory the Present Applications

The first task in quantifying the present workload requirements is to "inventory" the major applications running during the selected time-frame and to determine their associated *ADP requirements*. There are usually several sources of data available for performing this task, as discussed below. If an agency determines that no data (or inappropriate data) exists to accomplish this task, then a data collection effort must be undertaken consistent with the cost and expected benefit of such an effort.

2.1.1 Use of Accounting Log Data

The most common source of data about an application's *ADP requirements* is the system's accounting log. On some systems, a different accounting log exists for each type of processing mode. For example, on some systems, information for both batch and online applications are recorded in the same file; on other systems, a separate log file exists for online applications.

Because most accounting logs are designed for chargeback purposes, an account number or other organizational identifier can usually be associated with each application. As will be seen, these identifiers are useful in associating an application with its organizational function. Because an application is defined here to be what is generally termed a "job-step" or "online activity," applications running together as a "job" or "online session" will most likely have the same account number. Figure 1, for example, depicts three batch jobs, from two different accounts, each containing two applications (i.e., job-steps). Also depicted are two online activities under two different sessions.

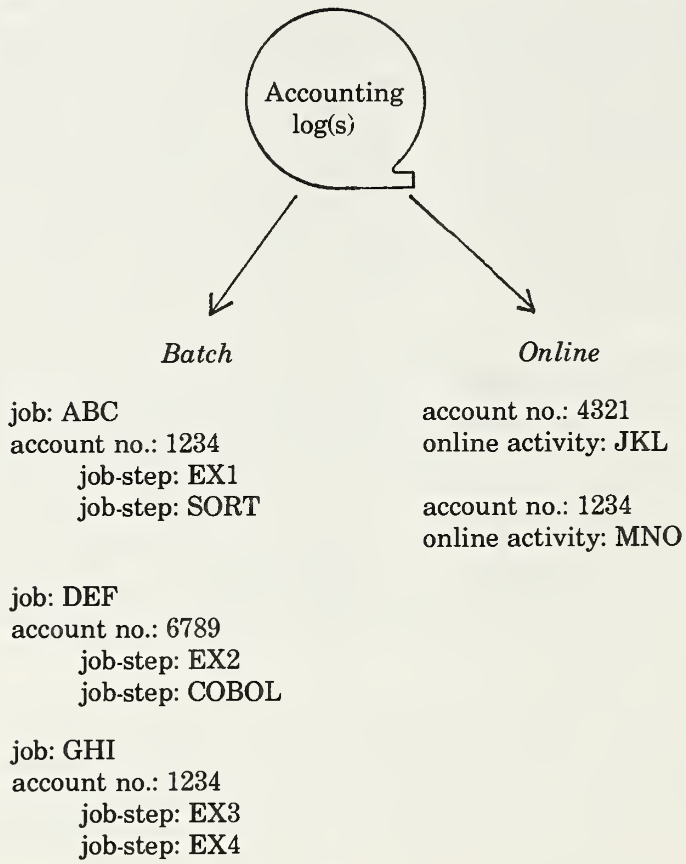

Figure 1. Analyzing accounting log data

An initial inventory of applications over the selected time-frame can be obtained by analyzing the accounting log associated with each processing mode. Such an inventory will consist of the following items for each application:

1. account number (or other organizational identifier);
2. application name; and
3. associated *ADP requirements* (see Table 1).

When it is not possible to associate a unique name with an application (item 2, above), the application can at least usually be generically named by class (e.g., "compile" or "execute"). A large part of item 3, an application's *ADP requirements*, can be obtained from the accounting log. Information that is commonly available includes:

1. resource usage data;
2. limited information on ADP operations performed (e.g., types of utilities invoked);
3. shift, priority, and security-related information; and
4. elapsed time in the system.

The accuracy with which the above informational items are collected and reported varies from system to system. Data reduction packages are usually available on most systems to analyze the accounting log file.

For online applications, some accounting logs provide additional information, such as:

1. response time (exclusive of external network delays);
2. terminal speed and type;
3. user wait time (i.e., the sum of user think time and typing time);
4. number of concurrent users for each application (especially for transaction-oriented applications);
5. application volumes (number and types of transactions processed); and
6. online commands executed.

Although most of the information in Table 1 can be obtained from the accounting log(s), other sources of data may also prove useful in quantifying an application's *ADP requirements*, as described below.

2.1.2 Use of Supplemental Sources of Data

a. Software Monitors

Software monitors can be used to obtain more detailed information on an individual application beyond that which appears in the accounting log [SVOBL 76]. The following is a partial list of data that can be obtained using a software monitor:

CPU usage,
channel activity,
memory used,
elapsed time,
number of transactions processed, and
data access by file.

Software monitors are resident in the computer system and collect data usually on a sampling basis. Software monitor data can be analyzed through the use of data reduction packages.

b. Hardware Monitors

Although hardware monitors are generally used to collect resource usage information on the entire system [CARLG 76], some are capable of obtaining data at the application level. Application-specific data that can be collected by hardware monitors usually includes CPU usage, channel activity, and paging characteristics. Response time monitors [NBS 78] are a special class of hardware monitors that can be used to obtain more accurate information on the response time associated with an online application. Communication line monitors can sometimes be used to obtain a sampling of terminal dialogues. As with software monitors, hardware monitor data can be analyzed by data reduction packages.

334-807 0 - 81 - 3

c. Online System File

In addition to its resource demands, an online application can also be described by the sequence and frequency of online commands executed. The following is a list of typical online commands:

delete a line;
list a file;
attach, delete, catalog, rename a file;
obtain status information;
queue a file for execution or printing; and
send or receive a message.

The terminal dialogue for an online session, that is, the sequence of online commands executed, can often be obtained, if not from the accounting log, then from a separate file associated with the online system. Care should be taken that the privacy and security aspects of analyzing and recording such information are properly considered. A terminal dialogue can then often be separated into logically-related sets of online commands (i.e., online activities). Additional information about online applications, such as (approximate) response time, terminal speed and type, number of concurrent users, and combined user think and typing time, can also be obtained from an analysis of such files. Online applications can also be described by a state transition diagram (see [WRIGL 76]) in which a "state" is an online command with accompanying probabilities of transitioning to other states (i.e., other types of online commands).

d. Miscellaneous

The identification of an application as either "vendor-supplied" or "user-written" (see Table 1) can often be obtained through a comparison of the application name with a file of vendor-supplied software (such as compilers, utilities, etc.). Information on the characteristics of data files used (see Table 1) can usually be obtained from the system's catalogs.

e. Hardcopy Output

One final source of information for an application's *ADP requirements* is the actual listings of user jobs. Through a manual analysis of selected job listings, the following data can usually be obtained for an application:

resource usage,
ADP operations performed,
elapsed time, and
file characteristics.

Collecting information from this source of data is a time-consuming process, and should be used only as a last resort.

2.2 Determine Application Groups

Having been inventoried and their associated *ADP requirements* quantified, applications may now be grouped by cost center, or other organizational identifier associated with the account number under which the application is run. Each such collection of applications will be termed an **application group** and represents a functional grouping of applications and associated present *ADP requirements* over the selected time-frame. Figure 2 depicts such a grouping for the inventory shown in Figure 1. Note that because an application group consists of applications with a common account number, it could contain one or more application systems.

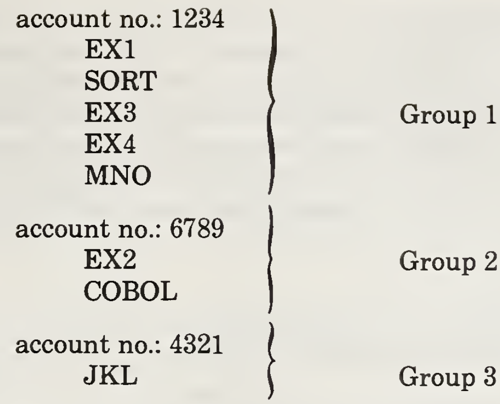

Figure 2. Sample application groups

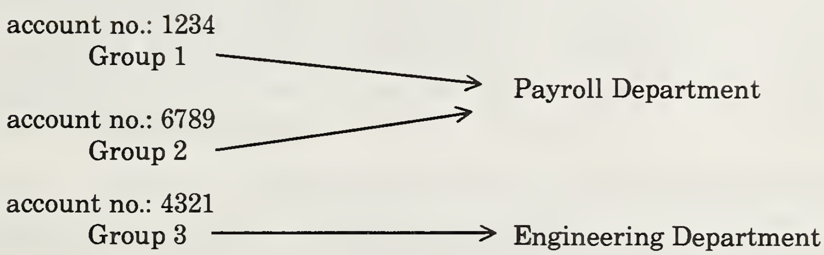

Figure 3. Mapping of application groups to organizational entities

2.3 Associate Application Groups with Organizational Entities

Because a mapping usually exists from the identifier associated with an application (e.g., account number) to an entity within the organization, each application group can next be associated with an organizational entity and function. For example, Figure 3 shows the mapping of the application groups in Figure 2 to specific organizational entities. By associating application groups with organizational entities, user surveys can then be conducted (in STEP 3) in order to obtain additional information on current application systems, as well as information on future requirements.

STEP 3. Survey Users

Prior to forecasting the aggregate workload, organizational entities can be surveyed in order to determine additional information about their present applications. In addition, surveys can be used to determine predicted changes to application systems—that is, changes to the *ADP requirements* of current applications, as well as the addition of new applications.

3.1 Obtain Additional Information about Present ADP Requirements

User surveys can be used to determine whether the initial inventory of applications obtained from STEP 2 is complete. If it is determined that there are important application systems that were not included in this initial inventory (for example, application systems that were processed in a time-frame other than the one examined in STEP 2), then information on the requirements for these application systems can be obtained from the sources of data discussed in STEP 2.

User surveys can be used to supplement information obtained from STEP 2. For example, as noted earlier, functional information about an application is generally not available in an automated form. However, user surveys can sometimes be used in conjunction with existing system documentation to obtain the following functional information about an application:

ADP operations performed,
frequency and schedule of production runs,
characteristics of source language programs,
file structures,
application dependencies, and
input/output volumes (e.g., checks generated).

In addition to the above, user surveys can be used to determine the required service levels for an application. As noted in STEP 1, the type of service requirement is determined by agency objectives and goals. The results of quantifying present, actual service levels can be compared with required service levels obtained from user surveys and can thus provide a quantitative base upon which management decisions can be made. Since the ability of the new computer system to provide the desired service may require additional system capabilities, the cost of providing this additional service versus the expected benefit to be derived should be evaluated.

3.2 Obtain Information about Future Requirements

When a user survey of future requirements is conducted, each organizational entity is furnished data on its present *ADP requirements* and asked to forecast, preferably in monthly increments, future changes to these requirements, as well as the addition of new applications or application systems. If present applications show seasonal changes and this tendency is expected in the future, then it is desirable to forecast on such a basis.

A change in future requirements can be caused by a number of factors; among them are changes in reporting cycles, changes in mission, and changes in budget. The users should attempt to identify the points in time when changes in requirements will occur.

When estimating their future requirements, the users should be asked how confident they are in the accuracy of their forecasts. The uncertainties in the users' future requirements estimates can then be used to bound the agency's aggregate future workload forecast (STEP 4).

In most instances, it is easier and sometimes preferable for the user community to forecast in user-defined units such as checks produced, loans approved, etc., rather than processing demands (i.e., ADP operations or resource usage). For stable applications, a relationship can sometimes be determined between a functional, quantifiable event and the processing demands associated with that event. Linear regression techniques can be used to obtain a linear relationship of this sort. When these techniques are used, the quantifiable event (e.g., number of paychecks) would be the independent variable and the processing demands (e.g., CPU time, I/O activity) would be the dependent variables. For example, linear regression techniques could be used to solve the following equation for A and B (the fitting constants), given many data pairs of the form (number of checks, CPU time):

$$\text{CPU time} = A \times \text{number of checks} + B.$$

Such a solution would represent the best fit of a line through the given pairs of data. Once this equation has been solved for A and B, future estimates (from the users) of "number of checks" can be used to forecast "CPU time." For example, based on known, future hiring goals, the organizational entity (in this case, the Payroll Department) can forecast the number of paychecks expected to be generated. This forecast can then be translated into a corresponding forecast in terms of processing demands. An analysis of residual values can be made to determine if certain basic assumptions are met with regard to the use of regression techniques. In addition to finding linear relationships, linear regression techniques can also be used to estimate nonlinear relationships through the use of logarithmic transformations [NEVIA 64].

In addition to forecasting expected future processing demands, each organizational entity should also identify changes in other *ADP requirements*, such as changes in application priority, security requirements, file structures, service times, and even changes in processing mode. A typical example of this is the application system that is presently batch-oriented, but will change to an online processing mode. In translating this batch application system into an online environment, several factors that will affect its future *ADP requirements* need to be considered. These include:

memory requirements,
number of transactions expected to and from the terminal,
number of characters transferred between the terminal and mainframe,
type of online activities,
file structure,
number and size of databases.

For example, consider an inventory control system that is currently running as a tape system in batch mode and is to be converted to a transaction-oriented environment. The volume of input and output data currently handled by the inventory control system should already have been quantified in STEP 2. At this point, the volume of data must be translated into the number of transactions and characters transferred to and from terminals. If there is a similarly designed application system running in an online environment on the present computer system, then the resource usage estimates can be derived from it.

ADP requirements for new application systems can be estimated based on already existing, similarly designed application systems. If there are no similarly designed application systems in existence, modeling techniques can be used, or manual estimates of resource usage can be made based on partial information of expected ADP operations for the new application system.

STEP 4. Forecast Future Workload Requirements

Because a benchmark will be constructed to test each proposed system's ability to meet future *ADP requirements*, it is critical that the aggregate workload to be processed throughout the expected life of the new system be estimated as closely as possible.

This quantification of the future workload should be done at least in yearly (preferably in monthly) increments in order to provide a quantitative base for determining potential augmentation points.

4.1 Forecast Aggregate Workload Requirements

Several approaches can be taken for projecting aggregate workload requirements:

1. extrapolate previous usage over the expected life of the new system;
2. estimate future *ADP requirements* from user surveys (STEP 3);
3. use a combination of approaches 1 and 2.

These three forecast approaches rely on two primary sources of data: historical data and user-supplied forecasts. Historical data consists of data accumulated over several previous months or years and is obtained from the same sources used to quantify the present workload (see STEP 2 and STEP 3.1). User forecasts over the short and long term are obtained from a survey of major organizational entities, as described in STEP 3, and are combined into an aggregate workload forecast.

The first approach ignores future changes (which might be presently known to the users) that will affect their future *ADP requirements*. Examples are known changes in budget or organizational function. In addition, it should be recognized that the growth and pattern of previous usage may have been constrained by the limited capacity or functional capabilities of the present system. These factors should be taken into consideration when a forecast is based on previous usage. While the second approach relies solely on user estimates, it sometimes tends to overestimate the actual, future requirements. The third approach consists of an extrapolation into future years based on the workload growth determined by a short-range user forecast (i.e., through year 2 or 3), as well as on the preceding years for which actual historical data exists. The least squares method or time-series analysis can be used for performing such an extrapolation. The technical team performing the benchmark construction is encouraged to elicit the assistance of a statistician familiar with these and other forecasting techniques.

It is sometimes useful to perform all three approaches and compare their respective workload forecasts. Figure 4 is an example of three forecasts obtained from each of the three approaches. Such a comparison produces upper and lower bounds within which the actual, future workload is likely to exist. Producing such bounds is useful so that an agency can weigh the benefits to be gained by a more accurate workload forecast against the costs required both to obtain it and to construct a benchmark to represent it. This will allow the agency to decide if the interval between the upper and lower bounds needs to be reduced through a more thorough workload forecast in order to reduce the risk of procuring an over- or under-sized system. It should be noted that when the bounds on the forecast are determined, the uncertainty in user estimates, as well as the errors inherent in the forecast techniques themselves must be taken into consideration. For example, a sensitivity analysis can be performed for varying ranges of user estimates in order to determine their effect on the growth of aggregate workload requirements.

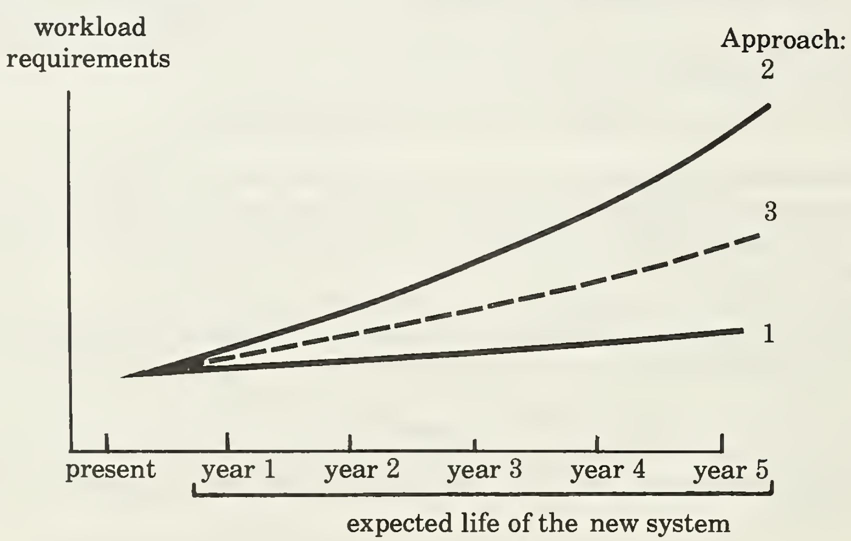

Figure 4. Three approaches to workload forecasting

4.2 Identify Potential Augmentation Points

Because of changing future workload requirements, the initial configuration of the new system may have to be augmented with additional capacity—such as CPU's, main memory modules, or input/output channels—at various points in time. As part of their proposals, prior to the actual live test demonstrations, the vendors are usually free to propose a configuration for each of these augmentation points, as well as for the initial configuration. Each new configuration proposed is generally benchmarked and is considered in the overall evaluation process.

Significant points in time which must be benchmarked and for which the vendor **may** propose a different configuration can be identified by the agency in the following manner. (These points in time will be referred to as "potential augmentation points," even though some vendors may elect not to propose an augmented system for these points in time.) Workload forecasts are analyzed to determine if significant changes in composition are expected to occur in the future workload at identified points in time. Figure 5, for example, depicts two significant changes in future workload requirements at augmentation points A and B (the y-axis in Figure 5 represents changing, aggregate workload requirements in terms of both character and volume). Such changes might be due to major, new workload components (e.g., a large online workload component is expected to go into production). Benchmarks would then be constructed by the Government to represent the workload associated with the end point of the initial configuration (point A in Figure 5), as well as the workloads associated with the end points of each augmentation interval (points B and C in Figure 5). The vendor is then usually free to propose a single configuration to meet the total workload, or a separate configuration for each potential augmentation (up to a Government-specified maximum).

If the workload is expected to increase uniformly in size over the total expected life of the new system, or over a significant period of time (as, for example, between points A and B in Figure 5), then the vendor is generally free to propose a separate configuration during these expected increases (again, up to a Government-specified maximum). The benchmark for each vendor-selected augmentation point is constructed usually by replicating benchmark problems or increasing data volumes in a Government-specified way.

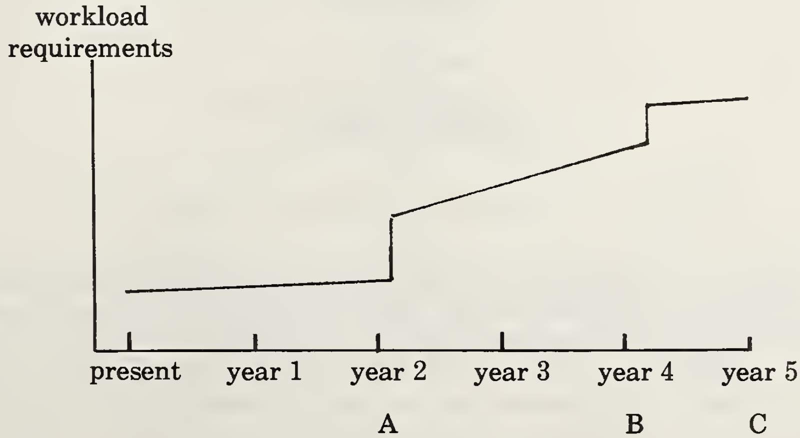

Figure 5. Sample augmentation points

A time period (e.g., a month) is selected at the end point of the initial configuration interval, as well as at the end points of potential augmentation intervals, and STEP'S 5 THROUGH 8 BELOW ARE REPEATED FOR EACH OF THESE SELECTED TIME PERIODS.

STEP 5. Categorize Future Workloads

Based on the information obtained from STEP 4, **workload categories** can be derived for each augmentation. A workload category is a collection of applications with similar *ADP requirements*. For example, Figure 6 depicts a hierarchical categorization of applications first by shift, then by processing mode, processing type, priority, and finally by processing demands. A workload category for user-written applications would consist of all applications run within the same shift, with the same processing mode/type, at the same priority, and with approximately the same processing demands (i.e., resource usage or ADP operations). Vendor-supplied software could be categorized by type (e.g., sort, compile, file maintenance).

A convenient technique for forming workload categories by processing demands is that of clustering. Cluster analysis techniques are useful tools for grouping many observations (in this case, applications). Each category (or cluster) represents the grouping of applications with similar parameter values (i.e., similar resource usage data or number and type of ADP operations performed). Clustering techniques vary widely in complexity, and range from "manual" clustering to statistical methods. Because most techniques frequently used in workload analysis can be considered special cases of clustering (histograms, for example, can be viewed as one-dimensional clusters), discussion is restricted here to the use of clustering techniques for forming workload categories.

The basis for any kind of cluster analysis is a set of clustering features (in this case, parameters describing ADP applications) that are important to the type of clustering being done. Feature selection techniques can be used to determine only the most significant parameters (see [AGRAAb 77]). Correlation analysis can next be performed on all remaining parameters to determine their interdependence. Those parameters that are completely dependent on other parameters can then be excluded.

shift
↓
processing mode
↓
processing type
↓
priority
↓
processing demands

Figure 6. Hierarchical categorization of applications

Statistical cluster analysis has been used for categorizing workloads both by resource usage and ADP operations, although, it has been used mostly for categorizing workloads by resource usage. Figure 7 depicts a workload clustering by the ADP operations "number of updates" and "number of records sorted." Note that, in practice, clustering techniques can be used to categorize the workload by a number of parameters, and not just two, as depicted in Figure 7. Each cluster in Figure 7 represents a different workload category, and each point within a cluster represents an application that performs the specified number of updates and records sorted. If the clustering is done by resource usage, then the analyst performing the

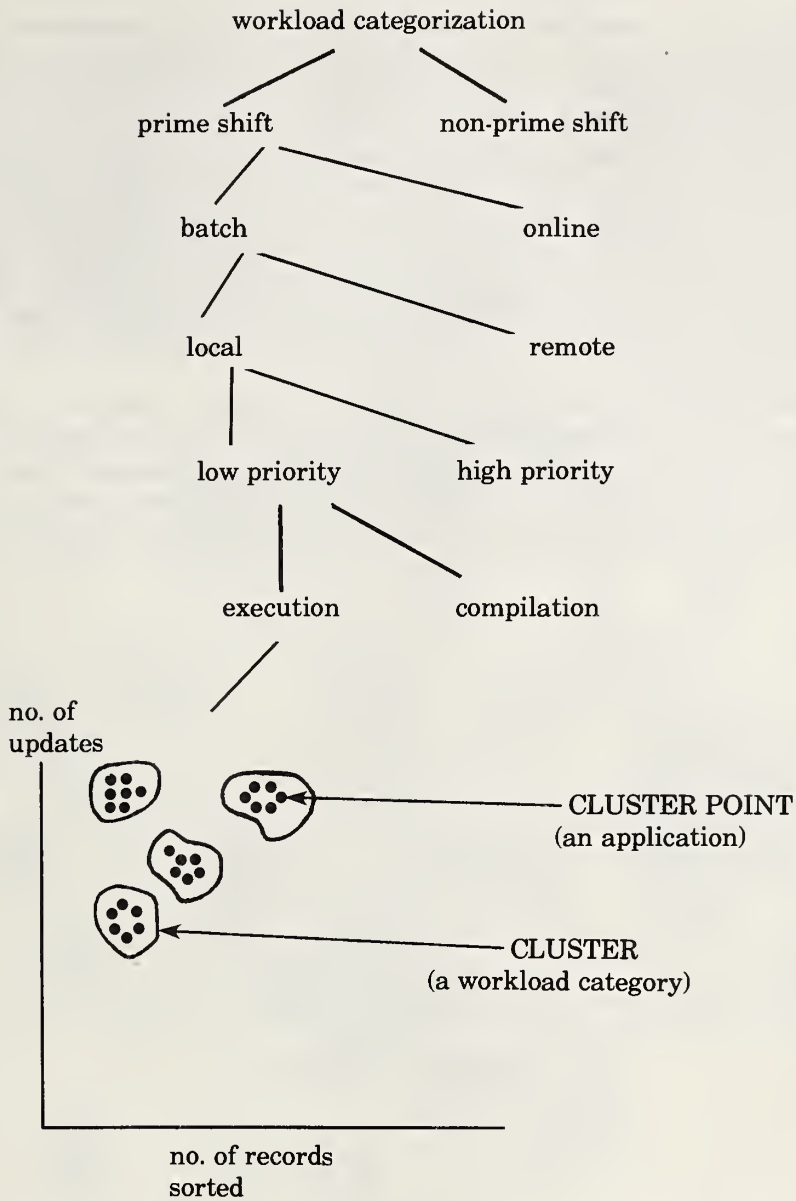

Figure 7. Workload clustering

statistical cluster analysis may determine whether the clusters represent some natural functional grouping of the workload by examining the applications in a given cluster; for example, all compile-only jobs might be found to reside in a low CPU, low I/O cluster. Detailed studies of various clustering techniques can be found in [AGRAAa 77] and [ARTIH 76].

Most cluster analysis techniques are sensitive to the range of parameter values chosen. Careful analysis should be made of the outliers in order to determine whether or not to eliminate them. It should be pointed out that outliers may in fact represent the highest resource-consuming jobs and their elimination may lead to a grossly inaccurate workload representation.

It should also be noted that the processing demands of an application must first be scaled before being used in a cluster analysis. This is true because an application is represented as a point in a multi-dimensional space and this point is used to calculate the distances between other applications represented also as points in this space. For example, if resource usage parameters are used to cluster the workload, then a difference of 5 seconds of CPU time between two applications cannot be compared to a difference of 250 I/O counts. In order to reduce the artificial dominance of some workload parameters, their values can be scaled, say, by making the average value of each workload parameter zero and the standard deviation one. It should also be noted that different clustering techniques, or even the same clustering technique employing different programming strategies, are likely to produce different results.

Various **ad hoc** techniques can be used to perform a "manual" clustering of applications when automated clustering techniques are not available. For example, Figure 8 depicts a graphical representation of a bivariate distribution of applications. This distribution is computed by dividing the range for a given parameter into a pre-determined number of intervals. The intersection of two

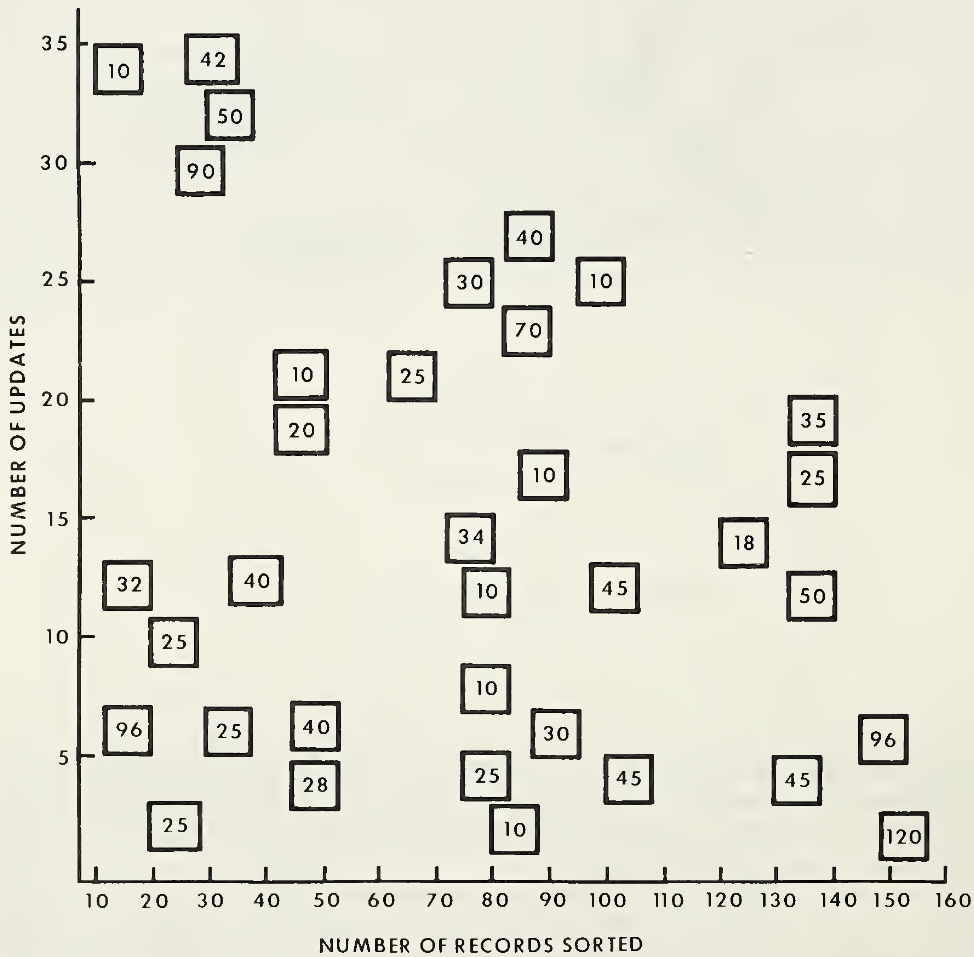

Figure 8. Sample bivariate distribution of applications

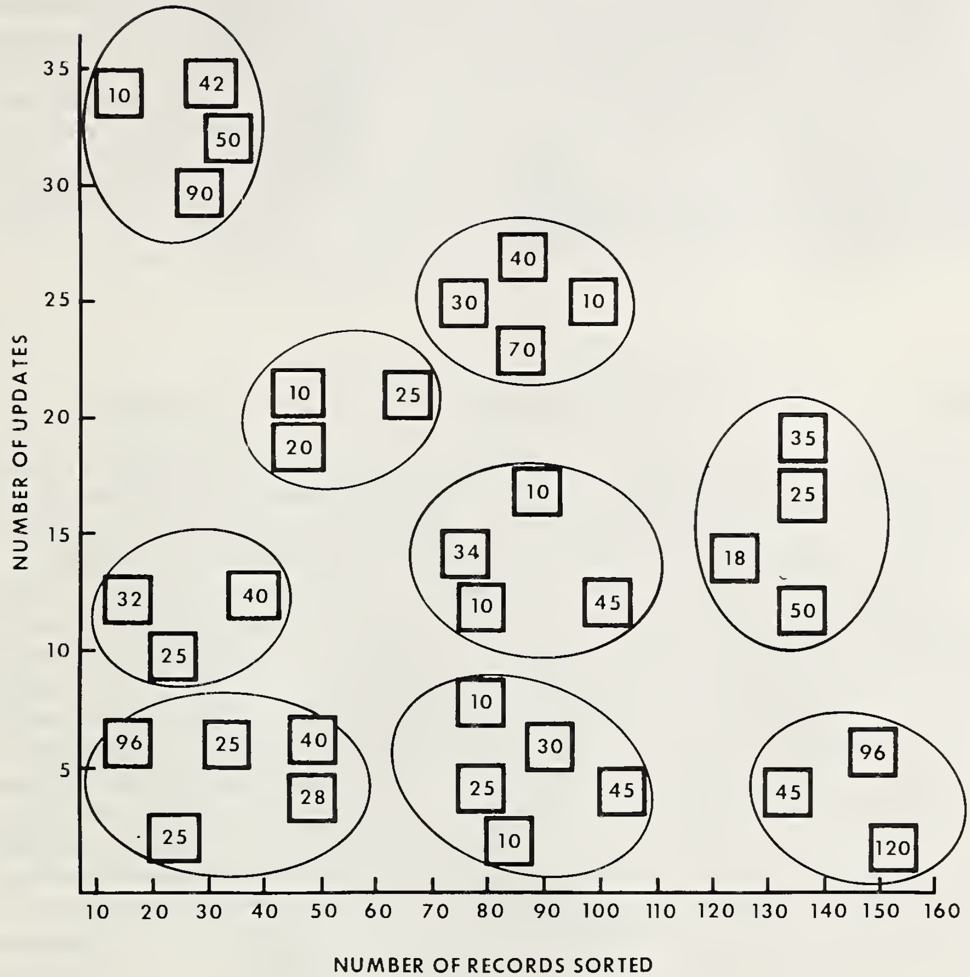

Figure 9. Manual clustering

intervals in a bivariate distribution becomes a two-dimensional "cell," and an application is placed in the appropriate cell based on its parameter values. The numbers in each cell in Figure 8 represent the number of applications falling in that cell. A collection of cells can then be "manually" (visually) clustered to represent workload categories, as depicted in Figure 9. For more than two features, however, this technique becomes cumbersome because several levels of bivariate distributions must be performed. For example, in order to add a third parameter to the two depicted in Figure 9, another distribution would be made where the y-axis contains the parameter pairs (no. of updates, no. of records sorted) for which corresponding cells exist in Figure 9, and the x-axis is divided into intervals for the additional third parameter. A cell in this second-level bivariate distribution thus represents all applications that have similar values for each of the three parameters.

A determination must be made at this point as to whether the various processing modes should be part of the same benchmark mix or whether they should each be represented in a separate benchmark mix. If, for example, batch and online workloads are expected to occur together on the new system, then the benchmark mix should represent both modes. On the other hand, if the workload to be processed during different shifts differs significantly in categories of work, then it may be desirable to represent the workload for each shift by different benchmark mixes. It is necessary to determine at this point the number of distinct workloads (and, hence, benchmark mixes) because the relative contribution and the scaling for each category (STEP's 6 through 7) ARE COMPUTED RELATIVE TO THE WORKLOAD WHICH THE CATEGORY BELONGS TO (e.g., prime or non-prime shift workload, in the case where each shift is to be represented by a separate benchmark).

STEP 6. Determine Relative Contribution of Each Category

The purpose of this step is to determine the relative contribution of each category (found in STEP 5) to the total workload it belongs to (e.g., prime or non-prime shift). This implies the need for a measure that represents the work or demands of an application (and by extension, all applications within a category).

A functional measure that is sometimes used to express the relative contribution of each workload category is the number of transactions processed by each application (assuming that several large applications exist with transactions requiring similar processing demands). In the more common case, however, involving an existing, heterogeneous workload, a measure quite often used to express the aggregate demands of an application is the system accounting unit (SAU) on the incumbent system. The generic term SAU will be used here to denote the accounting unit found on most third-generation computer systems; examples of such units are CRU, SUP, SRU, etc. Although the SAU is a system dependent measure (because it is computed based on resource measures) and sometimes even a processing-mode dependent measure (when it is computed differently for batch and online), it does represent, nevertheless, a reasonable approximation of the aggregate processing demands of an application, at least in a relative sense. In addition, the formula used to compute the SAU often attempts to approximate the stand-alone execution time of an application (which is itself a reasonable measure of the aggregate demands of an application, albeit also in system dependent terms).

Thus, one approach often used to determine the relative contribution of each category to the total workload it belongs to is to sum the SAU's for all applications within each category, and compute the ratio of this sum to the sum of all categories for the workload and time period in question. Figure 10 depicts the relative contribution of each of the workload categories depicted in Figure 7. Note that this use of an SAU requires that the projection of an application's *ADP requirements* also includes a projection of its SAU value. However, such a projection can often be obtained indirectly by using the projected resource usage data for an application to compute its SAU.

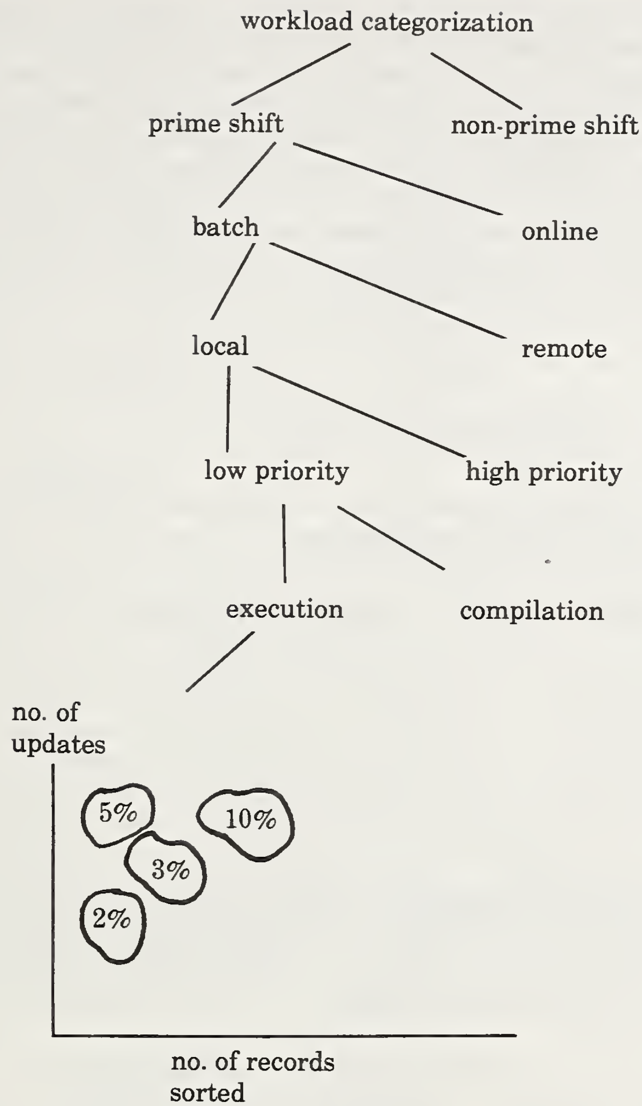

Figure 10. Relative contribution of workload categories

STEP 7. Scale Each Category

Because a benchmark problem may ultimately be constructed to represent each workload category, it is first necessary to determine the desired processing demands for each such problem. Intuitively, each benchmark problem should place demands on the system in proportionally the same way as does the category it represents. Results from STEP 6, together with the number of hours to be represented by the **complete** benchmark mix (usually 2 hours or less), can thus be used to determine the desired processing demands for each benchmark problem. Adjustments must first be made to the number of operational hours for the workload and time period represented by STEP 6. For example, changes in the number of shifts per week and consideration of maintenance and other overhead factors will impact the number of available operational hours (factors such as projected maintenance time should be computed as independently as possible from the agency's experience with its current system). For example, assume that a benchmark mix is to be constructed to represent the prime shift workload of the first month of year 2, whose processing demands were projected to be 12,500 SAU's and is estimated to contain 150 hours of operational time (after consideration for overhead factors, such as maintenance time, database backups, system initializations, etc.). Furthermore, assume that the distribution of SAU's over the month in question is uniform. If it is desired that the benchmark mix represent, say, 2 hours of elapsed time, then the number of SAU's to be generated by such a benchmark mix should be:

$$166.7 \text{ SAU's} = \frac{2 \text{ hrs.}}{150 \text{ operational hrs./month}} \times 12{,}500 \text{ SAU's/month.}$$

Now, assume that the categories for the prime shift workload during the month in question have the following relative contributions:

Workload Category	Relative Contribution
1	2%
2	5%
3	3%
4	10%
5	50%
6	20%
7	10%

When the percent of relative contribution for each category is applied to the total desired number of SAU's for the complete benchmark mix (in this case, 166.7 SAU's), it is found that the benchmark problems representing each of the categories should produce the following number of SAU's:

Workload Category	Desired Number of SAU's for Each Benchmark Problem
1	3.3
2	8.3
3	5.0
4	16.7
5	83.4
6	33.3
7	16.7
	166.7 SAU's

If the SAU's are not expected to be uniformly distributed over the time period in question, then some time-frame within the time period (e.g., the peak hour of the day) could be chosen, and the benchmark mix could be constructed to produce the corresponding number of SAU's.

In summary, the desired number of SAU's for each benchmark problem representing each category is determined as follows:

1. Determine the total number of operational hours for the workload and time period in question.

2. Using the workload projection in STEP 4, determine the total processing demands (usually in terms of SAU's) for the workload to be represented in this time period. If the SAU's are not uniformly distributed over the time period in question, then select a time-frame within the time period.

3. Determine the desired elapsed time (usually 2 hours or less) to be represented by the benchmark mix for the workload and time period in question.

4. Scale the total workload processing demands (for the workload and time period in question) by the ratio of desired elapsed time to total number of projected operational hours available for processing the workload.

5. Apply the relative contributions for each workload category determined in STEP 6 to the total processing demands for the benchmark mix (derived in 4., above) in order to determine the proper processing demands for each benchmark problem representing each workload category.

STEP 8. Represent Workload Categories with Benchmark Problems

The selection of benchmark problems to represent workload categories for the time period in question is the next step in the benchmark construction process. For workload categories that do not represent a significant relative contribution to the total workload, it may not be necessary to represent them with a benchmark problem. However, for some of these categories, which do represent applications that perform important functions, it may be necessary to include them in a separate, functional demonstration (see FIPS PUB 42-1 for additional guidance on constructing functional demonstrations). For example, such categories might include online graphics or utilities such as database restore, checkpoint/restart, etc. The relative contribution of the different modes of processing (batch and online) to the total workload must also be examined in order to determine if a mode of processing even needs to be represented. The next two sections of this step discuss the representation of batch and online categories, followed by a discussion on formulating each benchmark mix.

8.1 Represent Batch Workload Categories

Batch benchmark problems can be selected from either real programs or synthetic programs [MAMRS 79]. In either case, several important benchmark program characteristics should be considered.

The benchmark programs should be written in standard high level languages. The use of standard high level languages helps equalize each vendor's effort in implementing the benchmark and reduces both the agency's and vendors' conversion costs.

The benchmark program should represent all of the important *ADP requirements* of the workload category, including processing demands, file characteristics, etc. The service requirement that the benchmark program should meet when run on each vendor's system should be the same as that of the category it represents. Also, consideration should be given to representing specific operational requirements (such as multi-volume tape handling) that are believed to have a significant impact on system performance, and whose effects are known to vary extensively across systems. Furthermore, the benchmark problem should be scaled to reflect the proper amount of processing demands for the category, as determined from STEP 7. In some instances, it may not be possible to construct a benchmark problem that will "represent all of the important *ADP requirements* of the workload category" and still reflect the **scaled** processing demands of that category. This situation usually occurs for the smaller contributing categories. If this happens, either some less important *ADP requirement(s)* of the category is allowed not to be represented, or the benchmark problem is allowed to represent all of the category's

ADP requirements, and therefore result in a larger contribution of processing demands than was originally intended. Before either alternative is taken, the effects on the accuracy of the total benchmark and the objectives of the benchmark itself should be considered.

The benchmark programs should be as simple as possible without compromising their representativeness. That is, their logic should not be too complex, common programming techniques should be used, and good documentation should exist.

One way to select real programs representative of a cluster is to choose the one closest to the center point of the cluster. Of course, care must be taken so that other important aspects of the applications in the cluster, which were not part of the original clustering parameters (e.g., file structures), are not ignored.

Representing workload categories with real programs has its disadvantages, however. For example, security and privacy considerations may prevent the use of some of the programs and data files. It may be difficult to represent many different workload characteristics with a reasonable number of real programs. Also, real programs may be biased in favor of the incumbent vendor. It may be difficult to make them operational on various vendor systems because their interfaces to the operating system, database management system (DBMS), transaction subsystem, etc. may have to be modified. However, use of real programs has several advantages over synthetic programs, the primary one being better representation of the complexities of individual user jobs.

A benchmark mix consisting of synthetic programs that attempt to represent ADP operations (i.e., functionally-oriented synthetics) has certain advantages over one containing real programs. They are: data file conversion problems can often be eliminated through the use of a data generator; synthetic benchmark problems are easily modifiable and transportable; and, finally, they eliminate program and data security considerations often associated with real programs. However, synthetic programs tend to be stylized, and thus susceptible to optimizing compilers. Furthermore, the use of synthetic programs that attempt to represent resource usage characteristics (i.e., resource-oriented synthetics) is of questionable validity. The use of functionally-oriented synthetic programs is advantageous, however, when the projected workload is to contain new applications and real programs do not exist. Using information about similarly designed, existing applications, parameter-driven, functionally-oriented synthetic programs can be used to represent these new applications.

8.2 Represent Online Workload Categories

As with batch applications, the relative contribution of each online category to the workload in question is determined in STEP 7. Several factors need to be considered when selecting online activities to represent these categories. Such factors usually depend on the activities being examined. For example, for an edit session, the functions to be performed (such as move a line, delete a line, etc.), as well as the size of the edit files, must be considered when selecting representative online activities. If this analysis reveals that an actual session encompasses most of the activities representative of the online workload category, then this typical session may be used for generating online scenarios. If this is possible, a great deal of work can be avoided in constructing synthetic sessions.

Special consideration should be given if a DBMS is required on the new system, since each responding vendor may have a different database management package. A fair test of the vendor's proposed DBMS would be to measure the response time at the terminal for the user's transactions while the system is under an expected load. User DBMS functions (e.g., UPDATE) should be specified in vendor-independent terms. That is, a specific sequence and implementation of these commands (e.g., UPDATE: search, locate, modify) should not be imposed on the vendor. Other DBMS features (e.g., relational searches, DBMS backups and restores) may be included in a functional demonstration. Care should be taken that the appropriate number and types of DBMS functions are represented so that the vendor does not tailor the database in such a way as to reduce the DBMS processing unrealistically. Finally, it is desirable to use a data generator to create automatically the data used as input to the vendor's DBMS, instead of using actual data occupying several reels of tape.

For transaction-oriented applications, care should be taken in sequencing the input transactions. Rather than consisting of all transactions of the same type grouped together, the input stream for the benchmark should contain different types of transactions sequenced in as realistic a manner as possible. In addition, the arrival of transactions to the system under test—as sent, say, by a remote terminal emulator (RTE)—should not occur at regular intervals, but rather should be randomized to reflect more closely the actual, transaction-oriented environment.

8.3 Formulate the Benchmark Mix

After the benchmark problems that will represent each batch and online workload category have been determined, the next step is to create a benchmark mix for each workload during the time period in question (recall that STEP 5 may have determined that several benchmark mixes are desired for a particular potential augmentation point).

Although the benchmark problems should be chosen in such a way so as to be proportional to their corresponding workload categories, the manner in which they are replicated, sequenced, and combined is extremely important. Because the sequence in which jobs are submitted to a computer system may affect the running time of the mix, consideration should be given to the following. If the ADP installation implements manual job scheduling procedures as a matter of policy and that policy is expected to continue, then the benchmark programs should be sequenced in such a way so as to represent these policies. For example, job dependencies and priority requirements can be imposed on the benchmark programs in the same manner as their real workload counterparts. Benchmark programs representative of production and program development work can be intermingled if this is the operating policy of the installation. Or alternatively, production jobs might be sequenced together as a batch distinct from program development work.

In the usual case, however, the benchmark programs are arbitrarily sequenced, loaded into the input queue of the system under test, and the operating system is relied upon to decide automatically the proper sequencing of jobs, with minimal operator intervention. However, even in this situation, consideration must still be given to such items as job dependencies and priorities.

In the case of online activities, the configuration of active terminals must be decided. Such items as the type and number of concurrently active terminals, the configuration and types of communication lines, the sequencing of online tasks, and concurrency with batch work must be determined. See [GSA 79] for a detailed description of these and other items of concern.

In summary, the collection of benchmark problems for all categories represents the benchmark mix for the workload and time period in question. (Recall that STEP's 5 through 8 are to be repeated for each selected time period associated with the end points of the initial configuration and each of the potential augmentations, and each time period may result in more than one benchmark mix.)

STEP 9. Fine Tune Each Benchmark Mix on the Present System

Having formulated each benchmark mix, the team that will be in attendance during the vendor live test demonstrations should then exercise the timed benchmark tests on the present system. This is necessary in order to ensure that team members are familiar with their responsibilities and to help verify that the aggregate characteristics of each benchmark mix properly represent those of the projected workload it is intended to represent. For example, the number of iterations of programs and online activities or the volume of data might have to be adjusted in order to insure that each benchmark mix is representative of the intended workload in terms of:

1. resource consumption,
2. ADP operations,
3. number of concurrent terminals/users,
4. number of transactions processed per unit of time,
5. priorities, and
6. volume of data.

This information can be obtained from accounting logs, software monitors, hardware monitors, etc. The result of this examination might require adjustments to the appropriate benchmark problem(s).

When each benchmark mix is run on the present system, additional problems may arise for those benchmark mixes that contain a large online workload component and for which an RTE is to be used by the vendor. This is due to the fact that an agency will most likely not have an RTE at its disposal (this may also be true later, after delivery, if the benchmark is rerun prior to acceptance testing of the new system). If the agency does have an RTE, say in the form of an additional mainframe with supporting software and communications, then it should be used. When this is not possible, however, the agency should determine whether an internal "emulator" is available for its present system. Such emulators produce a controlled online load on the present system in one of two ways. Either the online benchmark activities are transmitted out to the front-end communications processor and back into the system (see Figure 11), or the work is transmitted internally to the online subsystem (see Figure 12). The former situation is preferred since the communications hardware and software are exercised. The disadvantage of using an internal emulator in either case, is the resultant overhead it imposes on the system and the difficulty in factoring out these effects. Some emulators do reside in the front-end processor, however, and the effects of overhead in these instances are reduced [WATKS 77].

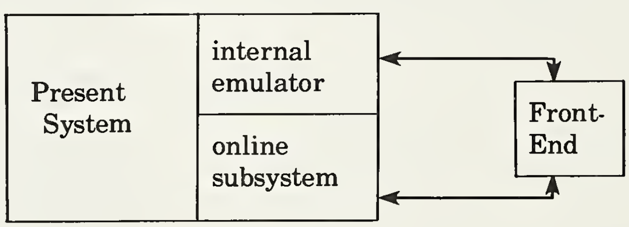

Figure 11. Use of a front-end and an internal emulator

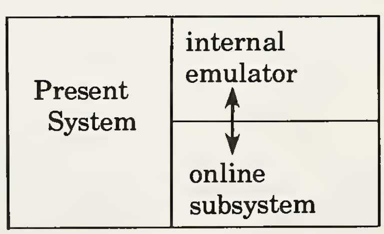

Figure 12. Use of an internal emulator

Benchmarks that require software not available on the present system may create an extra burden for personnel attempting to test the benchmark. That is, a benchmark may be developed in such a way that it depends upon software to be provided as part of the procurement (e.g., DBMS software). However, the benchmark components provided by the Government still must be tested. Software simulators can sometimes be developed to provide the missing functions.

STEP 10. Prepare the Benchmark Package and Test the Benchmark

This step is divided into two sections. The first section discusses the benchmark package; i.e., documentation of each benchmark mix, together with the documentation of the LTD rules. The second section discusses testing of the benchmark by running each benchmark mix on one or more systems other than the one on which it was developed. Because FIPS PUB 42-1 also provides guidance in both of these areas, as well as guidance in planning, conducting, managing, verifying, and evaluating the results of the benchmark, discussion will be limited here to topics not covered in FIPS PUB 42-1, or to an expansion of the more important topics that are in FIPS PUB 42-1.

10.1 Prepare the Benchmark Package

10.1.1 Document Each Benchmark Mix

A functional description of each benchmark problem, as well as internal documentation within each problem, should be provided in the benchmark package portion of the RFP. English-language scenarios for batch and online benchmark problems should be provided and, where possible, supplemented with sample scripts. Sample results of the benchmark, as well as the expected service time requirements for the benchmark problems, should be included as part of the benchmark package. A glossary of terms should also be provided to reduce any misunderstandings.

A general block-diagram showing the input files and their origin should be provided. For example, "file A generated by program ABC," "provided by the Government on tape 2," "vendor provided," "generated by data generator program XYZ" may be necessary qualifiers in such a description. The destination of the output files should be depicted on such a diagram. A description of each file should include information such as record length, blocking factor, number of records in the file, access method, storage media on which the file will reside when the benchmark is executed, field definitions, data formats, etc. The data provided to the vendors should be in a machine-independent format, and the volume of data provided on magnetic tape should be kept to a minimum. All data provided should be in compliance with Federal standards for media and interchange codes.

Constraints on modifications to the source code of benchmark problems must also be documented. Manual modifications beyond those necessary to interface with the vendor's system are normally not allowed. Source or object code optimization should be allowed only if the optimization mechanism will be part of the standard software delivered with the computer system (for example, the vendor's off-the-shelf optimizing compilers).

The RFP should require that each vendor meet with the agency benchmark team a few weeks before the LTD so that questions (on both sides) concerning the nature of the benchmark and the LTD can be resolved. Prior to such a meeting, the vendor should furnish the following information to the benchmark team:

1. a diagram of the complete configuration that is being proposed for each augmentation point, and the configuration(s) upon which the benchmark will be run (if different than proposed);

2. complete source program and data file listings, with a complete description of any modifications to benchmark programs or scenarios (including the exact changes made and reasons for the changes);

3. compilation listings for all programs showing job control information, compilation maps, size of the object modules, main (or virtual) memory allocations, disk or drum allocations, peripheral device requirements; also, complete listings of program outputs, and any other listings which would be a direct result of compilation and execution of the benchmark (e.g., diagnostics, cross-reference lists, etc.);

4. complete hardcopy of all operator/computer communications generated during compilation, loading, and execution of each benchmark problem;

5. listing of all software packages used to process the benchmark problems, including a list of all system generation routines and other system utilities that may be required (the software should be identified by release and version);

6. a complete set of manuals describing the system generation for each proposed configuration.

10.1.2 Document the LTD Rules

The rules for setting up and performing the LTD must be carefully documented in the RFP in order to avoid any misunderstandings between the vendors and the procuring agency. Furthermore, if not stated elsewhere in the RFP, the rules covering the following should also be stated:

1. allowable variations in the benchmark results;
2. acceptance and evaluation criteria of the benchmark results;
3. how the benchmark will be operated and supervised;
4. the environment during the benchmark (as discussed in more detail below).

a. Timed Benchmark Tests

When practical and only when it is believed necessary, the agency may require that the full complement of components be configured during the timed benchmark test, even if only partially used by the benchmark, in order to include the effects of device tables resident in memory, operating system overhead, file placement, channel contention, etc. (It should be noted that because such a requirement usually places an undue expense on the vendors and could limit the number of responding vendors, it should be stated only when absolutely necessary.) For example, the agency might require the vendor to configure a full complement of disks on which a set of "dummy" files might be loaded. The allocation of these files to specific disks should be done in the same manner as would occur for the real workload; namely, the vendor should have the system assign the files automatically, or the vendor should assign them manually using whatever utilities and suggested practices are contained in the vendor's user manuals. Care should be taken to prevent the vendor from physically arranging the data on or across disks in order to optimize only the benchmark. When it is not feasible to benchmark the complete proposed configuration, the agency may require the offeror to perform a functional demonstration for those devices or components that were not part of the timed benchmark test (see below).

The LTD itself must be well-documented. The allowable number and actions of operating personnel, which programs may be resident in memory, and execution constraints, if any, should all be clearly stated. The LTD documentation should also specify that the benchmark demonstrations must use the same versions and releases of the software and hardware as proposed by the vendor in response to the RFP, unless waivers are granted by the Government.

Pre-execution and start-up requirements must be documented. This should include items such as preloading of programs, files, databases, etc. prior to the timed test demonstration. When modifications will be made to the benchmark data files immediately prior to the test (in order to reduce the effects of any vendor tuning to a specific set of data), the procedures for doing so should be clearly specified.

Benchmark validation data requirements must be specified. That is, data should be requested which allows the benchmark team to verify the accuracy of results, as well as the correct performance of the benchmark. Sources for such data might include accounting logs, console logs, printer listings, RTE logs, and hardware and software monitor data.

b. Functional Demonstrations

Instructions for performing functional demonstrations must also be specified, if any are to be performed. Functional demonstrations are usually designed to test certain mandatory requirements or desirable features that cannot be satisfactorily evaluated from vendor proposals or would not be appropriate for inclusion in a timed benchmark test. Examples are data file security, utility capabilities, speed and capabilities of unit record equipment, and start-up and shut-down procedures. Component parts of the functional demonstration should be keyed to specific requirements in the RFP that the functional demonstration is designed to test. Furthermore, at least the following should be explicitly described: the material to be provided by the Government or vendor, what the Government expects to observe, and the criteria used to determine the acceptability of a given functional demonstration. The reader is referred to FIPS PUB 42-1 for additional guidance on conducting functional demonstrations.

10.1.3 Develop Internal Agency Documentation

In addition to developing the above external documentation which goes to the responding vendors, the agency should also maintain its own internal documentation on such items as the technical and policy decisions that were made which affected the benchmark construction, the data used to develop the workload forecasts, and the sources from which benchmark problems and data files were obtained. This information may prove useful later, especially over long acquisition periods when changes to the benchmark team are likely to occur.

10.2 Test the Benchmark

There are several reasons for running each benchmark mix on computer systems other than the current one, especially on systems similar to those likely to be proposed by the vendors. Running the mix on other systems can provide valuable information on the transportability of the benchmark problems from one vendor's system to an another. Doing so can also determine the correctness and clarity of both the benchmark mix and the supporting documentation. For example, errors introduced into a benchmark package commonly involve incorrectly generated benchmark tapes, incompatibilities between the benchmark problems and the accompanying documentation, inconsistencies in the documentation, and even program logic errors. It is likely that these and other errors will be detected if the benchmark mix is run on one or more other systems, especially if performed by personnel other than those who designed the mix. Running the mix on other systems is also useful for determining the repeatability of the benchmark problems by comparing the execution results to the results obtained on the present system. It is likely that the numerical precision will not be identical on different vendor systems, but it should be determined if the difference in results is due to execution errors or to numerical precision differences on other vendor systems.

It should be noted that some of the same problems associated with running the benchmark on the agency's current system may exist here also, notably, the need for a separate machine to function as an RTE and the need for transaction or DBMS software. For this reason, if the complete benchmark cannot be run on another system, at least significant portions of it should be run to test its transportability.

Running the benchmark on other systems has value, although limited, for validating the benchmark timing. It also gives some insight into the size of the systems likely to be bid.

SUMMARY

The previous sections have attempted to provide practical guidance on benchmark construction in a step-by-step fashion. Some agencies may find that the sequence of these steps is not suited to their particular needs; in this case, the Table of Contents can be used as a checklist. Others may find that some steps are already completed; e.g., users have already been surveyed as to their future requirements. In any case, this Guideline should be used as a basic reference guide on steps that should **at least be considered** when benchmarks are constructed for evaluating vendor offerings during the competitive acquisition of ADP systems.

It is hoped that this Guideline will help reduce agency costs for constructing benchmarks, vendor costs for implementing benchmarks, and agency risks in acquiring inappropriately sized systems.

GLOSSARY

ADP REQUIREMENTS—the complete description of an application (see Table 1).

APPLICATION—a logically distinct, identifiable problem presented to a computer system (e.g., a "job-step" or "online activity").

APPLICATION GROUP—a collection of applications having the same organizational identifier.

APPLICATION SYSTEM—a collection of related applications that perform a distinct agency function.

BENCHMARK—one or more benchmark mixes, together with the benchmark rules for running each mix.

BENCHMARK MIX—a set of benchmark problems that are properly combined to be representative of some future workload requirements.

BENCHMARK PROBLEM—a batch program or online activity that makes up a benchmark mix.

BENCHMARK RULES—the operational requirements associated with the running of a benchmark mix.

FUNCTIONAL DEMONSTRATION—use of a benchmark to test that a particular system has certain functional capabilities.

LIVE TEST DEMONSTRATION (LTD)—the actual running of a benchmark on a vendor's system as part of the evaluation process.

ONLINE ACTIVITY—a logical collection of online commands.

ONLINE COMMAND—a single command (such as "list") executed during an online activity.

ONLINE SESSION—the collection of all online activities performed between user logon and logoff.

POTENTIAL AUGMENTATION POINTS—significant points in time which must be benchmarked and for which the vendor is free to propose a different configuration.

PROCESSING DEMANDS—the ADP operations performed by an application, or the resources consumed by an application.

PROCESSING MODE—the manner in which an application is presented to a computer system for execution (batch or online).

PROCESSING TYPE—the particular way in which a processing mode is used (see Table 2).

SCENARIO—a vendor-independent description of a group of online workload demands to be performed during a benchmark mix, expressed as user functions.

SCRIPT—the set of instructions, data, and procedures that causes a particular RTE to impose specific online workload demands on a given system; includes both commands to control the RTE and the terminal dialogue.

SERVICE REQUIREMENTS—the timeliness requirements of an application (usually expressed as throughput, turnaround time, or response time).

SUPPORT REQUIREMENTS—global requirements needed to support the processing of an agency's total workload.

SYSTEM DESIGN CONCEPT—an idea expressed in terms of general performance, capabilities, and characteristics of hardware and software oriented either to operate or to be operated as an integrated whole in meeting a mission need [OMB 76].

TERMINAL DIALOGUE—a sequence of online commands executed during an online session.

TIMED BENCHMARK TEST—the use of a benchmark to test the ability of a system to meet certain service requirements.

WORKLOAD—a collection of agency applications.

WORKLOAD CATEGORY—a collection of applications with similar *ADP requirements*.

WORKLOAD REQUIREMENTS—the collection of *ADP requirements* for all applications in a workload.

REFERENCES AND SUGGESTED READINGS

[AGRAA 76] Agrawala, A. K.; Mohr, J. M.; Bryant, R. M. An Approach to the Workload Characterization Problem. Computer. 1976 June; 9(6): 18-32.

[AGRAAa 77] Agrawala, A. K.; Mohr, J. M. Some Results on the Clustering Approach to Workload Modelling. Conti, Dennis M.; Walkowicz, Josephine L., eds. Computer Performance Evaluation Users Group: CPEUG 13th Meeting; 1977 October 11-14; New Orleans, LA. Washington, DC: U.S. Government Printing Office: 23-38. (Stock no. 003-003-01839-2) (NBS SP 500-18).

[AGRAAb 77] Agrawala, A. K.; Mohr, J. M. The Relationship between the Pattern Recognition Problem and the Workload Characterization Problem. Morrison, Robert L., ed. Computer Performance Evaluation: Proceedings of the 1977 Sigmetrics/CMG VIII Conference on Computer Performance: Modeling, Measurement, and Management; 1977 November 29 to December 2; Washington, DC. New York, NY: ACM Order Dept.: 131-139.

[AGRAA 78] Agrawala, A. K.; Mohr, J. M. Predicting the Workload of a Computer System. Ghosh, Sakti P.; Liu, Leonard Y., eds. AFIPS Conference Proceedings: 1978 National Computer Conference; 1978 June 5-8; Anaheim, CA. Montvale, NJ: AFIPS Press: 465-471. Volume 47.

[ARTIH 76] Artis, H. Pat. A Technique for Determining the Capacity of a Computer System. Highland, Harold J., ed. Computer Performance Evaluation Users Group: CPEUG 12th Meeting; 1976 November 8-12; San Diego, CA. New York, NY: CPEUG; 1976: 151-162. Available from: NTIS, Springfield, VA; (PB 290701).

[BUCHW 69] Buchholz, W. A Synthetic Job for Measuring System Performance. IBM Systems Journal. 1969 April; 4: 309-318.

[CARLG 76] Carlson, G. A Guide to the Use of Hardware Monitors. EDP Performance Review. (Part 1) 1976 September; 4(9): 1-8. (Part 2) 1976 October; 4(10): 1-7.

[CONTD 76] Conti, Dennis M. Use of Synthetic Benchmarks for Estimating Service Bureau Processing Charges. Washington, DC: National Bureau of Standards (NBS); Institute for Computer Sciences and Technology (ICST). 1976 July. 47p. NBS Technical Note 920.

[CONTD 78] Conti, Dennis M. U.S. Efforts to Develop Standard Benchmark Programs. Ferrari, Domenico, ed. Proceedings of the International Conference on the Performance of Computer Installations (ICPCI 78): Evaluation and Management; 1978 June 22-23; Gardone Riviera, Lake Garda, Italy. Amsterdam, NY, Oxford: North-Holland Publishing Company; 1978: 55-66.

[CONTD 79] Conti, Dennis M. Findings of the Standard Benchmark Library Study Group. Washington, DC; National Bureau of Standards (NBS); Institute for Computer Sciences and Technology (ICST). 1979 January. 49p. NBS Special Publication 500-38.

[CROWE 72] Crowding, Edward F. A Controllable Synthetic Job Stream for Benchmarking. Poughkeepsie, NY: IBM Systems Development Division; 1972 June; TR 002173-1. 13p.

[DONAD 76] Department of the Navy/Automatic Data Processing Equipment Selection Office (DON/ADPESO). Handbook for Preparation of Vendor Benchmark Instructions. Washington, DC: DON/ADPESO, Software Development Division; 1976 October 29; 63p.

[DUBER 76] Dubes, Richard; Jain, Anil K. Clustering Techniques: The User's Dilemma. Pattern Recognition. 1976. 8: 247-260.

[FERRD 72] Ferrari, D. Workload Characterization and Selection in Computer Performance Management. Computer. 1972 July-August; 5(4): 18-24.

[FLEMP 79] Fleming, Patricia M.; Rucks, Andrew C. A FORTRAN Synthetic Program for Benchmarking. Weatherbee, James E., ed. Computer Performance Evaluation Users Group: CPEUG 15th Meeting; 1979 October 15-18; San Diego, CA. Washington, DC: U.S. Government Printing Office: 193-199. (Stock no. 003-003-02118-1) (NBS SP 500-52).

[GILBD 78] Gilbert, Dennis M.; Mulford, James O.; Spiegel, Mitchell G. Guidance for Sizing ADP Systems (ADPS's). Weatherbee, James E., ed. Computer Performance Evaluation Users Group: CPEUG 14th Meeting; 1978 October 24-27; Boston, MA. Washington, DC: U.S. Government Printing Office: 305-330. (Stock no. 003-003-01985-2) (NBS SP 500-41).

[GSA 79] General Services Administration (GSA) Automated Data and Telecommunications Service. Use and Specifications of Remote Terminal Emulation in ADP System Acquisitions. Washington, DC: GSA/ADTS (CDD); 1979 August; FPR 1-4.11. 102p.

[HOWAP 76] Howard, Phillip C., ed. Evaluation and Comparison of Software Monitors. EDP Performance Review. 1976 February; 4(2): 1-9.

[HUGHH 78] Hughes, Herman D. Workload Characterization of Computer Systems. Wright, Linda S.; Buzen, Jeffrey P., eds. Proceedings of the CMG IX International Conference on Management and Evaluation of Computer Performance; 1978 December 5-8; San Francisco, CA. Bethesda, MD: Computer Measurement Group, Inc.; 1978: 81-93.

[JOHNL 76] Johnson, L. Arnold. Characterization of an Interactive Time Sharing Workload. Highland, Harold J., ed. Computer Performance Evaluation Users Group: CPEUG 12th Meeting; 1976 November 8-12; San Diego, CA. New York, NY: CPEUG; 1976: 107-112. Available from: NTIS, Springfield, VA; (Document no. PB 290-701).

[JOHNL 77] Johnson, L. Arnold. Validation—All Important in Benchmarking. Conti, Dennis M.; Walkowicz, Josephine L., eds. Computer Performance Evaluation Users Group: CPEUG 13th Meeting; 1977 October 11-14; New Orleans, LA. Washington, DC: U.S. Government Printing Office: 71-74. (Stock no. 003-003-01839-2) (NBS SP 500-18).

[KERNB 73] Kernighan, B. W.; Hamilton, A. Synthetically Generated Performance Test Loads for Operating Systems. ACM Proceedings of SIGME Symposium 73; 1973 February 26-28; Palo Alto, CA. New York: Association for Computing Machinery (ACM); 1973: 121-126.

[KINGC 74] King, Carl. User Develops Procedure To Generate Synthetic Benchmark Jobstream. EDP Performance Review. 1974 April; 2(4): 1-5.

[MAMRS 79] Mamrak, S. A.; Abrams, M. D. A Taxonomy for Valid Test Workload Generation. Computer. 1979 December: 60-65.

[MCNEJ 77] McNeece, J. E.; Sobecki, R. J. Functional Workload Characterization. Conti, Dennis M.; Walkowicz, Josephine L., eds. Computer Performance Evaluation Users Group: CPEUG 13th Meeting; 1977 October 11-14; New Orleans, LA. Washington, DC: U.S. Government Printing Office: 13-21. (Stock no. 003-003-01839-2) (NBS SP 500-18).

[MORGD 73] Morgan, D. E.; Campbell, J. A. An Answer to a User's Plea? ACM Proceedings of SIGME Symposium 73; 1973 February 26-28; Palo Alto, CA. New York: Association for Computing Machinery (ACM); 1973: 112-120.

[MUKHA 76] Mukherjee, A.; Lauro, J. An Improved Benchmark Performance Evaluation Technique for Vendor Selection Studies. Highland, Harold, ed. Computer Performance Evaluation Users Group: CPEUG 12th Meeting; 1976 November 8-12; San Diego, CA. New York, NY: CPEUG; 1976: 197-201. Available from: NTIS, Springfield, VA; (Document no. PB 290-710).

[NBS 77] National Bureau of Standards. Federal Information Processing Standards (FIPS) Publication. Guidelines for Benchmarking ADP Systems in the Competitive Procurement Environment. Washington, DC: NBS; 1977 May 15; FIPS PUB 42-1. 28p.

[NBS 78] National Bureau of Standards. Federal Information Processing Standards (FIPS) Publication. Guidelines for the Measurement of Interactive Computer Service Response Time and Turnaround Time. Washington, DC: NBS; 1978 August 1; FIPS PUB 57. 26p.

[NBS 80] National Bureau of Standards. Federal Information Processing Standards (FIPS) Publication. Guidelines for the Measurement of Remote Batch Computer Service. Washington, DC: NBS. 1980 May 1; FIPS PUB 72. 39p.

[NEVIA 64] Neville, Adam M.; Kennedy, John B. Basic Statistical Methods for Engineers and Scientists. Scranton, PA: International Textbook Company; 1964. 325p. LCCCN: 64-23494.

[NUTTG 76] Nutt, Gary J. Tutorial: Computer System Monitors. ACM Sigmetrics: Performance Evaluation Review. 1976 January; 5(1): 41-51.

[OLIVP 74] Oliver, Paul; Baird, George; Cook, Margaret; Johnson, Arnold; Hoyt, Patrick. An Experiment in the Use of Synthetic Programs for System Benchmarking. AFIPS Conference Proceedings: 1974 National Computer Conference and Exposition; 1974 May 6-10; Chicago, IL. Montvale, NJ; AFIPS Press; 1974: 431-438. Volume 43.

[OMB 76] Office of Management and Budget (OMB). Circular No. A-109. Washington, DC: OMB; 1976 April 5. 12p.

[SHETA 74] Shetler, A. C.; Bell, T. E. Computer Performance Analysis: Controlled Testing. Santa Monica, CA: The Rand Corporation; 1974 April; R-1436-DCA. 84p.

[SVOBL 76] Svobodova, Liba. Computer Performance Measurement and Evaluation Methods: Analysis and Applications. New York, NY: American Elsevier Publishing Company, Inc.; 1976. 143p.

[TIMME 73] Timmreck, E. M. Computer Selection Methodology. Computing Surveys. 1973 December; 5(4): 199-221.

[WATER 77] Waters, Robert E. Selection of ADPS for the Air Force Academy: A Case Study. Conti, Dennis M.; Walkowicz, Josephine L., eds. Computer Performance Evaluation Users Group: CPEUG 13th Meeting; 1977 October 11-14; New Orleans, LA. Washington, DC: U.S. Government Printing Office: 71-74. (Stock no. 003-003-01839-2) (NBS SP 500-18).

[WATKS 77] Watkins, S. W.; Abrams, M. D. Survey of Remote Terminal Emulators. Washington, DC: U.S. Government Printing Office; 1977 April. 80p. (Stock no. 003-003-01756-6) (NBS SP 500-4).

[WRIGL 76] Wright, Linda S.; Burnette, William A. An Approach to Evaluating Time Sharing Systems: MH-TSS A Case Study. ACM Sigmetrics: Performance Evaluation Review. 1976 January; 5(1): 8-28,37.

U.S. GOVERNMENT PRINTING OFFICE : 1981 O - 334-807

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $13; foreign $16.25. Single copy, $3 domestic; $3.75 foreign.

NOTE: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

DIMENSIONS/NBS—This monthly magazine is published to inform scientists, engineers, business and industry leaders, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing. Annual subscription: domestic $11; foreign $13.75.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).

NOTE: The principal publication outlet for the foregoing data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Services, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services, Springfield, VA 22161, in paper copy or microfiche form.

U.S. DEPARTMENT OF COMMERCE
National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

OFFICIAL BUSINESS

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-211

3rd Class Bulk Rate

U.S.MAIL