DOCUMENT RESUME

ED 179 206 IR 007 868

AUTHOR Conti, Dennis M.
TITLE Computer Science and Technology: Findings of the Standard Benchmark Library Study Group. Final Report.
INSTITUTION National Bureau of Standards (DCC), Washington, D.C.
REPORT NO NBS-SP-500-38
PUB DATE Jan 79
NOTE 57p.
AVAILABLE FROM Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402 (Stock No. 003-003-02009-5, $2.40)

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS *Computer Programs; *Computer Science; *Cost Effectiveness; Federal Government; *Information Systems; *On Line Systems; Private Agencies; Program Evaluation

ABSTRACT

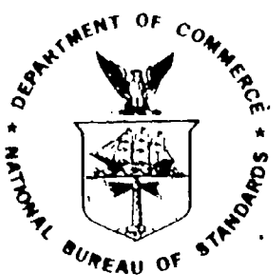
This report presents the findings of a joint government/industry study group which investigated the technical feasibility of standard benchmark programs for testing vendor systems in the competitive selection of computer systems within both private industry and the federal government. As part of its investigation, the study group reviewed earlier efforts to develop and use such programs on the part of the Department of Defense, the Auerbach Corporation, H. Lucas, the Mitre Corporation, and the Department of Agriculture (USDA). Several issues dealing with the implementation, maintenance, cost-benefit, and acceptability of standard benchmarks emerged as a result of this review. The problems encountered by the study group, notably the lack of an accepted definition of "representativeness," prevented it from arriving at a definitive statement of feasibility. However, several areas that were identified as topics requiring further investigation are presented in this report. A list of references, a glossary of terms, the USDA mapping procedure, and sample evaluation criteria are appended. (Author/FM)

COMPUTER SCIENCE & TECHNOLOGY:

Findings of the Standard Benchmark Library Study Group

Dennis M. Conti

Institute for Computer Sciences and Technology,
National Bureau of Standards
Washington, D.C. 20234



U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Jordan J. Baruch, Assistant Secretary for Science and Technology

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued January 1979

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

National Bureau of Standards Special Publication 500-38

Nat. Bur. Stand. (U.S.) Spec. Publ. 500-38, 57 pages (Jan. 1979)
CODEN: XNBSAV

Library of Congress Cataloging in Publication Data

Conti, Dennis M.
Findings of the standard benchmark library study group.
(Computer science & technology) (NBS special publication; 500-38)
Supt. of Docs. no.: C13.10:500-38
1. Electronic digital computers—Evaluation. I. Title. II. Series. III. Series: United States. National Bureau of Standards. Special publication; 500-38.
QC100.U57 no. 500-38 [QA76.9.E94] 602'.1s [001.6'4] 78-606168

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1979

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
Stock No. 003-003-02009-5 Price $2.40
(Add 25 percent additional for other than U.S. mailing).

ACKNOWLEDGMENTS

The author wishes to thank Mr. Terry Potter of the Digital Equipment Corporation (formerly with Bell Telephone Laboratories) and Mr. Norris Goff of the U.S. Department of Agriculture for their participation as study group members and as contributors to this report.

TABLE OF CONTENTS

FINDINGS OF THE STANDARD BENCHMARK LIBRARY STUDY GROUP

by

Dennis M. Conti

ABSTRACT

This report presents the findings of a Government-industry study group investigating the technical feasibility of standard benchmark programs. As part of its investigation, the study group reviewed earlier efforts to develop and use standard benchmark programs. Several issues dealing with the implementation, maintenance, cost/benefit, and acceptability of standard benchmarks emerged as a result of this review. The problems encountered by the study group, notably the lack of an accepted definition of "representativeness," prevented it from arriving at a definitive statement on feasibility. However, several areas were identified as topics requiring further investigation and are presented in this report.

Key words: Benchmarking; benchmark library; selection of ADP systems; standard benchmarks; synthetic benchmarks; workload characterization; workload definition.

1. Introduction

Benchmarking is an accepted mechanism for testing vendor systems in the competitive selection of computer systems within both private industry and the Federal Government. However, due to the rising cost of benchmarking on the part of both agencies and vendors, new methods need to be explored that will help reduce the overall costs of benchmarking. For this reason, the concept of "standard" benchmark programs has received renewed interest. A collection (or "library") of such programs could serve as a source from which agencies would select parameterized, functional synthetic programs to supplement their normal benchmark mix. In this context, a "functional synthetic program" is a computer program which is written to perform some pre-defined ADP function. Several important questions

remain, however, related to the feasibility of such an approach.

A Government-industry study group was formed in 1976 to determine the technical feasibility of the standard benchmark library concept. This report first surveys past efforts to develop and use standard benchmarks, and then summarizes the problems encountered by the study group. The report ends with a set of conclusions and suggestions for future work.

1.1. Background

Government-wide concern for benchmarking-related problems has been evident since at least 1969 when it was a major topic at the Conference on the Selection and Procurement of Computer Systems by the Federal Government, sponsored by the Office of Management and Budget.

In December 1972 the Commission on Government Procurement issued the following recommendation (Recommendation D-14) to the Executive Branch [14]:

> "Develop and issue a set of standard programs to be used as benchmarks for evaluating vendor ADPE (automatic data processing equipment) proposals."

In response to this recommendation, the General Services Administration initiated and chaired a committee of Executive Branch agencies which included the National Bureau of Standards (NBS), the Department of Defense, the Veterans Administration, the National Aeronautics and Space Administration, and the (then) Atomic Energy Commission. The committee developed an Executive Branch position paper dated March 27, 1974 [3] which stated that:

> "The feasibility of developing and issuing a set of standard programs to be used as benchmarks throughout the Federal Government for evaluating vendor ADPE proposals has not yet been established. If it is determined that these benchmarks are feasible, it is the recommendation of this committee that the recommendation be adopted by the Executive Branch as stated by the Commission on Government Procurement."

The Executive Branch position paper added that:

> "The primary objective of Recommendation D-14 was perceived to provide a mechanism to reduce the costs incurred by both the user and computer vendor in the benchmark process."

It also stated that:

"...much preliminary work needs to be done to test the feasibility of various approaches to standard benchmarks."

The position paper also pointed out that "criteria had not yet been established for determining feasibility" and that such criteria should be established "at an early date."

In May 1976, the Office of Management and Budget gave notice in the Federal Register of acceptance of Recommendation D-14 on behalf of the Executive Branch, and assigned lead agency responsibility to NBS as part of its existing central management role and ongoing efforts in benchmarking. NBS was directed to "coordinate and seek advancements in benchmarking within the executive branch" and to "publish various guidelines and documents, as appropriate."

Shortly before this time, NBS began a cooperative study effort with participation from the U.S. Department of Agriculture and Bell Laboratories to examine the technical feasibility of the development and use of functional synthetic programs as a basis for a common-use ("standard benchmark") library, one of several possible approaches responsive to Recommendation D-14. All three of these organizations had extensive experience in the development and use of synthetic benchmark programs.

1.2. Perspective

The technique of benchmarking remains a necessary and important tool in the competitive evaluation and selection of computer systems within both private industry and the Federal Government. This is true for several reasons. It is acceptable to the computer industry as a fair and unbiased live test of a vendor's proposed system. It is a mechanism by which an agency can model its current and projected workloads in such a way as to ensure that the vendor's proposed system will be of an appropriate size. It is a test mechanism which is repeatable within acceptable limits from one vendor to the next. Finally, for most batch benchmarks, the benchmark can be run against the newly installed system as part of an agency's acceptance testing procedures.

Benchmarking as currently practiced within the Federal Government usually consists of five distinct phases. During Phase 1, the workload to be performed by the new system is defined. This usually requires an analysis of the current

workload, a prediction of its future growth, and an estimate of new applications. In Phase 2, a benchmark is constructed to represent the defined workload, often in terms of some critical period of the workload (e.g., a peak month). During Phase 3, the benchmark is tested, sometimes by running it on a system other than the agency's current one. The benchmark is then modified to eliminate any errors or major machine dependencies, and is suitably documented for vendor use. In Phase 4, each competing vendor makes necessary and allowable changes to the benchmark in order for it to run on his system. Each vendor also undertakes to configure a system capable of processing the benchmark within some agency-determined time constraints. Finally, in Phase 5, the benchmark is run as part of a timed live test demonstration, and its performance is compared against the agency-defined constraints. During each of these phases, a cost is incurred either by the agency (Phases 1, 2, 3), by the vendor (Phase 4), or by both the agency and the vendor (Phase 5). The impact of the benchmark library concept on each of these costs is discussed in Section 3.3.

Although benchmarking is an important sizing tool, it is not an exact one. Benchmark runs are approximations to true workload demands over some agency-determined time frame. The degree to which a benchmark is representative of the true workload depends upon the complexity of the real workload, the accuracy with which future workload demands can be predicted, and the amount of effort the agency invests in the workload definition and benchmark construction phases. Producing high-quality benchmarks is usually a very expensive process for an agency. Low-quality benchmarks, on the other hand, are less expensive to produce, but usually result in higher costs to the vendors (as in the case of poorly documented programs), in addition to a higher risk that the procured system may not adequately satisfy the agency's requirements. It is the need for high-quality benchmarks at less cost to both the agencies and vendors that has prompted various efforts to establish a library of standard benchmark programs.

2. Previous Efforts

Several early efforts, notably those within the Department of Defense, attempted to address the benchmark library concept. Other related works include the use of standard benchmark problems by the Auerbach Corporation, a paper by Lucas in 1972 in which he outlined a set of modules that could be used to construct a functional benchmark, and

a study by the Mitre Corporation in 1975 in which results of a limited test of the benchmark library concept were presented. More recently, the use of an internal set of standard benchmark programs by the U.S. Department of Agriculture in their own procurements appears to be the most promising effort toward establishing feasibility. Each of these activities is discussed in more detail below.

2.1. Department of Defense Efforts

a. Air Force efforts

In 1971, a study conducted for the U.S. Air Force by the Mitre Corporation [11] resulted in a plan for a standard benchmark library for use in the competitive selection of computer systems by the Air Force Directorate of Automatic Data Processing Equipment Selection (MCS). The study included a feasibility study and an economic analysis of the standard benchmark approach as it applied to Air Force procurements. The study outlined the objectives and operation of a benchmark library, and presented several issues related to its use. Among the issues raised were:

1. Could vendor systems evolve in such a way that they would eventually be "tuned" to process the standard benchmark programs in a manner more efficient than the real workload?

2. What form should the benchmarks take (e.g., actual user programs vs. small CPU and I/O (synthetic) modules)?

3. Can users build representative workload models (i.e., benchmarks) using standard benchmark programs?

This last point was determined to be "the single most important issue in consideration of an MCS standard benchmark library." Because of this, it was suggested that "a trial run of the use of library programs to specify system workloads should be performed before the library concept is fully implemented." The study also estimated the level of staff and computer resources needed to implement the library, in addition to the dollar savings to the Air Force based on its use. Because the investment decision would "just about break even" (i.e., costs would equal benefits), it was concluded that the decision whether to implement the library should be based on non-monetary

benefits, such as reduced time to complete a procurement and reduced vendor costs. However, the study added that the most critical problem was whether user workloads could be represented by benchmark programs chosen from a standard library, and that this question could only be answered through experience. The study called for an early review of feasibility and a test run of the library as soon as it became operational. Apparently no further work was undertaken on this effort.

b. Army efforts

The development of standard benchmarks within the Department of the Army began in September 1972 in response to recommendations made by a Department of Defense (DOD) task force investigating the time and cost of ADPE procurements. This development effort became a full-time project within the U.S. Army Computer Systems Support and Evaluation Command (USACSSEC), although the project was coordinated by a joint steering committee composed of members from the Army, Navy, Air Force, and Defense Supply Agency. Initial efforts centered on the development of functional synthetic benchmark programs, data files to be used by the synthetic programs, and the development of a set of procedures for the use, distribution, and maintenance of the programs.

A Contributor's Symposium on Standard Benchmarks was held at USACSSEC in October 1972 for the purpose of refining the standard benchmark concept. Participants at the symposium included representatives from ADPE vendors, the (then) Business Equipment Manufacturer's Association (BEMA), interested universities, ADPE research firms, and the joint steering committee. The following excerpt from Department of the Army Pamphlet No. 18-10-2 [1] summarizes the results of this meeting:

> "The symposium was keyed to the 'utility' of standard benchmarks, using the Steering Committee's concept as a 'strawman.' The symposium was successful in meeting the established goals and in familiarizing many of the potential users with this concept."

The USACSSEC effort resulted in a contract with Galler Associates to "define a 'standard benchmark' and its usage." Although the Galler contract culminated in an extensive report [4] describing a "kernel" approach to the standard benchmark concept, the USACSSEC nevertheless felt that there were still several unanswered questions and unresolved problems. Among these was the problem of mapping user

workload requirements into the proper set of "kernels." This appears to be the extent of the USACSSEC effort.

c. Navy efforts

A related effort was begun in June 1973 within the Department of the Navy's ADPE Selection Office (ADPESO). This effort, partly in support of the DOD effort and partly for in-house use, was directed toward developing a small set of synthetic programs which could be used to "enhance an existing set of natural benchmarks in order to gauge specific system characteristics" [2]. Although the Navy effort produced five synthetic benchmark programs in which parameters could be set to force a prescribed load on various system resources (e.g., the CPU, I/O), several difficulties were reported. Among them were the dependency of the parameters on the nature of the system being evaluated, and the "sheer magnitude of the number of combinations of program parameter values" [13]. The study concluded that although synthetic programs could be controlled to produce a prescribed processing load on a given system, it was not possible "to arrive at a generalized, comprehensive, and accurate model of system workloads except in the most trivial cases." It added, however, that "if one accepts a 'modest' workload characterization, aimed more at reflecting extremities and crucial areas rather than comprehensiveness, it is possible and reasonable to construct a benchmark from a set of synthetic modules." No further work has been reported on this effort.

2.2. Auerbach Standard Benchmarks

Perhaps the earliest reported use of standard benchmarks was by the Auerbach Corporation in the development of their Standard EDP Reports [6,7]. These standard "benchmarks" were actually problems that covered a number of commonly performed ADP functions, such as file updating. The problems were hand-coded in assembly language for each vendor's system. Published instruction times were then used to calculate stand-alone problem time. A number of standard equipment configurations were defined to make comparative vendor evaluations easier. Execution times were estimated for each problem on each configuration. Users had to relate their special needs to these standard problems, and, because they were coded in assembly language, the problems were written differently for each vendor's system. The problems were not actually run on vendor systems, and

the estimated execution times did not consider multi-programming effects. This approach has apparently not been used since approximately 1971.

2.3. Lucas Modules

In a 1972 article [9], H. Lucas suggested that "a set of industry-wide synthetic modules be developed and provided by each computer manufacturer for his equipment." The intended use of these modules was primarily to assist users in modeling their workload (i.e., constructing a benchmark) for use in the selection process.

The proposed synthetic modules were divided into three categories: compiler attributes, operating system attributes, and program execution. Both the compiler and operating system categories contained modules primarily concerned with evaluating error detection features. The program execution category attempted to "represent all of the common operations found in both commercial and scientific data processing." Examples of such execution modules are: fixed point operations, stress analysis, forecasting model, and fixed length record update. Each of these proposed modules had associated with it one or more adjustable, but very general, parameters. Sample parameters included: number of calculations and precision, size of problem, number of forecasts and number of periods, and number and size of fields updated.

Although Lucas suggested that a user could construct a benchmark by selecting a group of synthetic modules from such a collection, he did not specifically address the problem of how this mapping from user requirements into synthetic modules and parameter settings should be done. He simply states that "the evaluator must determine the anticipated job load for the system to be evaluated" and that "he then selects representative models (i.e., synthetic modules) and joins them together into jobs which model that load."

2.4. Mitre Study

A study conducted by the Mitre Corporation in 1975 [8] for NBS stated three primary objectives: "to develop the Application Benchmark Library concept, to perform a preliminary feasibility test of this concept and to identify related areas for further study." The "development of the

concept" consisted of a suggested approach concerning the structure, creation, use, maintenance, and documentation of an application library. The "preliminary test" consisted of a controlled testing of two parameterized application programs, one written in FORTRAN and the other in COBOL. "Areas for further study" included investigations into the "operational" and "economic" feasibility of the library concept. One of the suggested "operational feasibility" tests included testing the ability to map user programs into library programs. In summary, the Mitre report suggested a physical structure for the library, outlined a library maintenance procedure, and showed that the resource demands of parameterized programs could be controlled in a predictable manner.

2.5. Department of Agriculture Experience

In 1972-1973, as part of its procurement of a new system for which few operational programs existed, the U.S. Department of Agriculture (USDA) undertook to develop a set of functional synthetic benchmark programs. Although the procurement was subsequently consolidated with other procurements, the same benchmark programs, with revised workload estimates, were used for this consolidated procurement. Three vendors submitted proposals, and all three demonstrated their proposed systems using the synthetic benchmarks. The consolidated procurement was cancelled, however, without an award being made. At the present time, USDA is going forward with several new, independent computer procurements. Each procurement involves sizing the present and future workloads of a different group of USDA agencies. The same basic set of synthetic benchmark programs used in the original consolidated procurement is being used for several of these procurements [10]. However, the programs have been upgraded in a number of ways since they were first developed. More importantly, a standard procedure was developed by USDA for its agencies to follow in projecting their workloads and mapping them to the synthetic programs. The following paragraphs discuss the USDA benchmark programs, the workload mapping procedures, and various technical considerations and issues related to the USDA effort.

a. Structure of the programs

Each of the USDA benchmark programs is designed to perform some common data-manipulation function. Major categories of functions are: (1) batch versus on-line

processing, (2) serial versus non-serial data accessing, and (3) data retrieval versus data update operations. A synthetic program was developed to represent each required combination of these major categories (for example, "batch serial update"). This effort resulted in a set of synthetic programs which represent distinct ADP operations across many applications, rather than programs which represent complete applications (such as "payroll"). The synthetic programs are inherently quite small and generate little CPU load except for that associated with moving transactions and data records in and out of memory. A common routine is incorporated into each program, however, which can be set to consume any amount of CPU time and any amount of memory. All on-line synthetic programs are designed to execute in conjunction with vendor-supplied transaction processing software, which is expected to pass to the programs one transaction at a time on a demand basis.

The synthetic programs are supported by a number of other software and procedural components, which together constitute a benchmarking system. These supporting programs include: a data generation program, a post-demonstration analysis program, a workload mapping procedure, and a workload tally program. Some of these components are relevant to this report and are therefore discussed at greater length in the following paragraphs.

b. Technical considerations

By virtue of its use in actual procurements, the USDA benchmark system has had the benefit of several critical, technical reviews. The more salient technical issues of the USDA standard benchmark effort are discussed here.

First, it has been proven feasible to map the workloads of a variety of USDA agencies to the benchmark programs. This issue is discussed at greater length in the next section. The USDA mapping effort did result in one or more new synthetic programs, or variations of programs, being proposed in order to more closely match certain major workload functions. Each proposal for a new program was evaluated to determine whether the resulting improvement in representativeness would be sufficient to justify the cost of developing the new program. On occasion, new programs were deemed to be necessary.

There was considerable concern at the outset of the USDA effort whether a vendor could take unfair advantage of some inherent characteristic of the synthetic programs --

for example, by placing the entire executable portion of code in a small, high-speed memory. The approach USDA took in dealing with this issue was to attempt to identify each potential weakness and correct it. A technical solution was developed for each potential weakness that was identified. USDA reports that no weaknesses have since been found which could not be overcome.

One major problem which USDA faced was interfacing their benchmark programs with sophisticated vendor software for which standards did not exist. Although this issue is not peculiar to synthetic programs, it is nevertheless important enough to mention here. In particular, the USDA benchmark depends upon a transaction processor and a data-base management system. However, only the most fundamental functions of these subsystems are used and even then, the vendors are allowed to modify the program interfaces. Although a more accurate workload representation could be produced if segments of the benchmark programs were tailored to the vendor software, this was not deemed feasible for a number of reasons. One major reason, presumably, was the desire to run the same, unmodified programs on all vendor systems.

One potential weakness of standard benchmark programs, referred to in Section 3.1 of this report, is the potential for the programs to influence the evolution of vendor systems. Nothing in the USDA experience can provide an answer one way or another on this issue.

c. Workload mapping

Because the current series of USDA procurements involve several different USDA agencies whose computer processing is performed at various computer centers, each agency is required to project its own future workload to be supported by the new systems. Technical personnel supporting the procurements, however, do provide the discipline to assure the compatibility of format, in addition to combining the workload projections for each center.

Early in its procurement efforts, USDA deemed it necessary to use a standard procedure for mapping agency workloads to the synthetic benchmark programs. Such a procedure was developed and has since evolved as personnel of several USDA agencies have used it. The workload mapping procedure is incorporated into this report as Appendix B. In summary, the procedure consists of four steps:

1. Identify major agency functions that result in an ADP workload. Where practical, functions are budget line items, such as "cotton loans." Establish a discrete unit of activity measure for each function (e.g., "number of loans").

2. Determine what ADP operations result from one occurrence of each function. These ADP operations are further quantified in terms of occurrences of various synthetic benchmark programs, or other specific benchmark workloads, such as program compilations.

3. Project the units of activity for each major agency function over the system life. Where practical, this activity is performed by budget personnel or other non-ADP persons.

4. Extend the quantifications of agency functions to ADP operations; i.e., to synthetic programs and other benchmark components. USDA has developed a computer program to assist its agencies in performing this step.

Step 2 above appears to be the most tedious, and requires that personnel have a thorough knowledge of their ADP operations. These personnel must also be thoroughly familiar with the synthetic benchmark programs. USDA reports that approximately eight hours of tutoring are required to familiarize personnel with these procedures. Further discussions are sometimes necessary to clear up any misunderstandings that may surface later. Nevertheless, it is reported that agency personnel, without prior knowledge of the benchmarking system, have performed the mapping process effectively, and in several instances, with relatively little training. This training procedure has been the source of some changes to the synthetic programs, since it is here that new people have the opportunity to review the programs and surface deficiencies with respect to the way the programs represent real ADP operations.

d. Effectiveness

The USDA benchmarking system appears to be satisfying its three major objectives.

First, a single procedure and a single set of tools and programs are serving to benchmark a series of systems. Repetitive use of the same tools will certainly result in

much better calibration and much less cost to the Government than would the development of a new benchmark for each procurement. It is premature to claim similar cost savings for the vendors, but it seems likely that their subsequent benchmark costs using these programs will be reduced.

Second, in order to equalize their proposals, all vendors are provided with the same demonstrable workload. The fact that the original, albeit aborted, procurement resulted in three demonstrated and proposed systems indicates that this objective was achieved. The three vendors who benchmarked in this early procurement effort did not report any suspected biases in the synthetic programs. In fact, a bias was claimed in one of the few operational programs which were included in the benchmark. USDA reports that recent analysis of vendor proposals and benchmark results (which cannot be published for proprietary reasons) indicates that the three responding bidders were as close in their configurations as could be established by such comparisons.

The third USDA objective was to assure that the systems which are proposed have the proper capacity to perform the projected workload. Strictly speaking, the only way to prove that this objective is achieved is to track the installed system's ability to meet the workload demands over the system's life. This assumes of course that the workload projections can be accurately made. As a practical matter, there are a number of other ways that the confidence level in the "correctness" of these benchmarks can be improved. Steps which USDA has taken include simulation, analytical analysis, and extensive execution of the benchmark on multiple systems. Some of these efforts have led to a more careful analysis of different elements of the benchmark and, in certain instances, have resulted in various adjustments to the benchmark programs themselves. In general, this analysis has supported, to the extent possible, the validity of the USDA benchmarks.

3. The Benchmark Library Study Group

Because it was assumed that enough work had previously been done to determine the feasibility of a standard benchmark library, an NBS-sponsored study group was formed in 1976 to address this question. As will be seen, this assumption proved false, principally because there existed neither within private industry nor within the Government any accepted criteria for determining when a benchmark was

"representative" of a computer workload.

The study group consisted of personnel from the Department of Agriculture, Bell Laboratories, and NBS. It met several times between March 22, 1976 and October 13, 1976. The stated objective of the study group was to "...attempt to establish the technical feasibility of benchmark library concepts for use within the Federal Government." In order to accomplish this objective, the following tasks were established:

1. Define relevant terms.

2. Determine scope of the benchmark library.

3. Identify potential problems associated with the benchmark library concept via interviews and a detailed review of previous efforts.

4. Determine criteria against which a proposed benchmark library can be evaluated for the purpose of determining its acceptability. Although evaluation criteria should be established for four major areas (technical, management, cost, and acceptability), emphasis was to be placed on the technical aspects.

5. Apply the evaluation criteria established above to existing or proposed benchmark library prototypes.

6. Based on the above results, determine, in general, whether any benchmark library (existing or proposed) is technically feasible (i.e., adequately satisfies the established evaluation criteria).

Task 1 resulted in a glossary of terms (see Appendix A). As a result of Task 2, the following scope was defined:

> "The study will address the feasibility of establishing and maintaining a library of synthetic application programs which will be useful for inclusion in benchmarks. More specifically, it will be limited to programs with these characteristics:
>
> (a) They may be written in standard COBOL or FORTRAN and must contain only standard components of those languages.
>
> (b) They are capable of representing batch or on-line transaction-processing applications

primarily of a 'commercial' (vs. 'scientific') nature which are describable by well-defined functions."

The results of Task 3 are described below. It soon became apparent as a result of Task 4 that determining the technical feasibility of a library of standard benchmark programs required much more preliminary work than had already been done. Section 4 of this report discusses this problem in more detail, and Section 5 suggests future courses of action.

Several issues evolved during the course of this study relative to the implementation, maintenance, cost, and acceptability of a library of standard benchmark programs. The following paragraphs briefly discuss each of these issues and attempt to assess their impact on the overall feasibility of standard benchmark programs.

3.1. Implementation Issues

a. Identification of a set of ADP functions common to many agencies

Central to the standard benchmark concept is the assumption that there exists a reasonably small number of ADP functions common to many agencies. Before a benchmark library could be developed, it would thus be necessary to first identify these functions. This could be accomplished either by surveying large Government installations or by reviewing the processing and benchmark requirements found in recent computer system Request for Proposals (RFP's). Assuming such a set of functions exists and can be identified, then benchmark programs could be written or obtained to implement these ADP functions. It is this collection of benchmark programs which would constitute the benchmark library.

b. Ability of benchmark programs to accurately represent agency workloads

Given that a set of common agency functions can be identified, a related, but equally important question, remains: Can the benchmark programs which implement these functions be combined and parameterized in such a way as to accurately represent agency workloads? For example, it may be found that many agencies perform a particular type of sort function. Although a benchmark program could be

written to duplicate this function, the question remains whether the program can be parameterized to adequately account for differing agency volumes, file structures, etc. This problem is further complicated by the lack of an accepted definition of what it means for a benchmark to be "representative" of a workload.

c. Synthetic programs could produce "overwhelming side effects"

A suggested alternative to the "functional" benchmark programs as described above is the use of resource-oriented synthetics. These synthetics are parameterized programs which, can be controlled to place a prescribed load on major system resources. The resource-oriented synthetics perform no useful work, but rather they exercise selected system resources in some pre-defined manner, for example looping on a series of CPU-bound statements. One of the problems that has been raised relative to the use of resource-oriented synthetics as standard benchmark programs is their inability to represent a given workload's resource demands across system lines [13]. For example, because they are usually written in a higher-level language, the translation of certain language constructs, such as a PERFORM statement in COBOL or a DO statement in FORTRAN, may produce such drastically different resource demands from system to system, that the synthetic's ability to represent the real workload is destroyed.

d. Unknown effects of optimizing compilers on "stylized" synthetic programs

Another problem that has been raised relative to the use of resource-oriented synthetic programs concerns the unknown, uncontrolled effects of optimizing compilers [13]. Because they are highly "stylized" (i.e., artificial in nature), such synthetic programs may be more (or less) susceptible to the effects of optimizing compilers. Consequently, the resulting performance impact on the synthetic programs may not be typical of that which would occur to the real workload. This problem also applies to some extent to functional benchmark programs.

e. Possibility of inherent biases for or against some vendors

A problem related to the use of any set of standard benchmark programs concerns the possibility of inherent program biases for or against some vendors. Although a

benchmark should place a representative load on each vendor's system, the benchmark should not perform actions above and beyond those needed to represent the actual workload. If it does, the benchmark may unduly bias one vendor over another.

A suggested solution to this problem is the incorporation of some mechanism, as part of the library's normal maintenance procedures, for responding to and resolving vendor complaints. Such actions may consist of eliminating questionable programs from the library, or modifying them to the satisfaction of all vendors.

f. Possible evolution of vendor systems tailored to benchmark programs

Assuming that a library of benchmark programs is usable, the question has been raised whether vendor systems will evolve in such a way as to maximize the performance of the benchmark programs, at the expense of the workloads which will actually run on those systems. Some continuous mechanism would therefore be needed, again as part of the library's normal maintenance procedures, to monitor the possible development of this situation.

g. Inability of synthetic programs to adequately test compilers, operating system control features, etc.

Finally, because of the limited number of programs that might be in a benchmark library, there is the danger that such system functions as compiler diagnostic procedures, operating system utilities, etc. would not be adequately tested. However, as suggested by Lucas [9], standard programs for testing these features could be developed.

3.2. Maintenance Issues

a. Ability of benchmark programs to meet state-of-the-art changes

Because of the highly dynamic nature of computer architectures and languages, a library of benchmark programs would have to be adequately maintained in order to prevent them from becoming obsolete. Obsolescence may result either because the programs would simply no longer run, or because they would be incapable of representing important, new architectural features. This latter point is exemplified by the recent popularity of paging systems: a benchmark

program not capable of representing the pattern of memory references of a functional application could be biased either in favor of or against some vendors. These potential problems, of course, also apply to current benchmark methods. In order to keep the benchmark programs consistent with state-of-the-art architecture and language features, an on-going review of the benchmark library programs would be needed.

b. Mechanisms needed to resolve agency and vendor problems and complaints

Irrespective of the particular benchmark programs in the library, no set of programs will satisfy all agency needs. Also, it is possible that a vendor may claim that one or more of the programs is biased for or against a particular system. Prompt resolution of these problems requires a maintenance mechanism capable of extending the library if enough agencies find it deficient in particular functional areas, and of objectively testing vendor claims of bias.

3.3. Cost/Benefit Evaluation

As input to an overall feasibility study of the benchmark library concept, the cost of such a library, in relation to its expected dollar benefits, should be evaluated. If a library of standard benchmarks were developed, agencies would have access to well-documented programs, easily portable across vendor lines, with which to construct or supplement their normal benchmark mix. This would result in reduced time and cost to agencies in constructing and documenting their benchmarks, as well as a reduction in vendor conversion costs. In addition, well-documented and tested benchmarks would most likely also reduce the time to complete a live test demonstration, a cost savings to both agencies and vendors. In a full cost/benefit evaluation, these benefits should be weighed against the cost to develop, use, and maintain a library of standard benchmarks. The benchmark study group did not conduct such a cost/benefit analysis other than to identify the above factors.

3.4. Acceptability to Agencies and Vendors

As part of a general feasibility study on the benchmark library concept, the anticipated use of the programs by agencies would have to be evaluated. This could be accomplished, as an example, by offering a preliminary set to a number of agencies conducting procurements and evaluating their use of the benchmark programs. It should be pointed out that several procurements have already taken place in which agencies have used pre-existing benchmark programs because they were available, well-documented, and fairly representative in function.

In addition to evaluating agency acceptance, vendor response to the standard benchmark concept should be solicited. It is anticipated that some vendors will welcome clean, well-documented programs as a way of reducing their benchmarking costs. As stated in the Executive Branch position paper on Recommendation D-14, "CBEMA's (Computer and Business Equipment Manufacturers Association's) primary concern ... is to insure that benchmarks take a form such that they can be constructed to be representative of the user's needs, to be consistently representative across vendor equipment lines, and not to restrict the vendor's ability and responsibility to configure his computer systems for most efficient processing." The vendor community has in the past cooperated with Federal efforts to arrive at better benchmarking approaches (a good example of this is the joint Government-Industry Remote Terminal Emulation Project [5]). There is no reason to believe that vendors would not cooperate in addressing the standard benchmark concept.

4. Problems Encountered in Attempting to Determine Technical Feasibility

In attempting to answer only the technical feasibility question (and not such other related questions as cost/benefits, acceptability, etc.), the benchmark library study group determined that a set of evaluation criteria should be established. Using these criteria, a candidate benchmark library could then be objectively evaluated as to its technical acceptability. These criteria were to be established apart from any particular benchmark library.

As a result of a concerted effort to establish such evaluation criteria, it was soon determined that there was no common agreement among the study group members (or for that matter, within the ADP community as a whole) concerning

the meaning of "representativeness" as it applies to benchmarks of existing workloads. Since the representativeness question was central to the evaluation criteria, this raised an obvious obstacle.

For discussion purposes, a theoretical approach was developed for defining "representativeness." A series of experimental tests (i.e., "evaluation criteria") were proposed such that if a candidate benchmark library "passed" these tests, then it would be deemed "technically acceptable," at least as far as its "useability" and "portability" are concerned (see Appendix A for a definition of these terms). This process is outlined in Appendix C and is an example of the type and complexity of evaluation criteria which the study group envisioned. It was generally agreed, however, that current benchmarking practices are not subjected to this level of rigorous definition and that such a degree of representativeness may not be achievable in practice. This did point out the need, however, for an empirical and acceptable test of representativeness.

Finally, in attempting to determine technical feasibility, the question arose whether the standard benchmark approach should be compared against existing benchmark construction approaches or whether it should be examined on its own merits. Since more traditional approaches to benchmarking have themselves never come under close, scientific scrutiny, it was believed that the benchmark library concept should be evaluated relative to existing practices.

5. Conclusions

Based on the previous findings, the benchmark library study group concluded that although the standard benchmark library concept has been used with apparent success within particular agencies (e.g., USDA), there is not yet sufficient data to establish the feasibility of such an approach for Government-wide use. The continued use of such an approach by USDA, however, and their post-installation experiences will provide more useful data to help answer some of the issues and problems raised earlier. Furthermore, the use of USDA's benchmarks by other agencies on an ad hoc basis will also provide valuable experiential data to help further answer the feasibility question as it applies across agency lines. To this end, NBS is currently exploring with USDA the possibility of making the USDA benchmark programs, along with a companion user's guide,

available to all Federal agencies. If this is done, the benchmark material would be distributed through a central source, such as the National Technical Information Service (NTIS). Requests for the benchmark material could then be monitored as an indicator of agency interest in the standard benchmark concept.

As a result of the study group's review, it was apparent that a technical foundation had not yet been established for addressing several fundamental questions in all phases of the benchmark process: workload definition, benchmark construction, etc. It was also clear that the best of known practices [12] are being used by only a handful of agencies. Furthermore, in spite of the relatively large number of Government procurements that have been conducted thus far, surprisingly little data exists on the relative effectiveness of alternative benchmark approaches to properly size computer systems. Some specific questions that the study group believes should be addressed are:

1. What should be the objectives, constraints, and quality measures of a benchmark mix demonstration?

2. Does there exist a common set of ADP functions across agencies?

3. Can a benchmark program be parameterized in such a way so as to accurately represent these logical functions, as well as any agency-required data volumes?

4. How can possible benchmark biases be identified and eliminated?

5. What are the proper analysis techniques which should be used to define a workload prior to benchmark construction?

6. What is the proper definition of "representativeness" in the competitive selection environment?

In addition to answering the above questions, more of an exchange of ideas and experiences is needed among agencies who have conducted computer system procurements. Furthermore, in keeping with the spirit of Recommendation D-14, other approaches to reducing benchmarking costs should

also be explored. One example is the development of a "library of tools" to assist agencies in the workload analysis and benchmark preparation phases. It is believed that only through an in-depth analysis of the problems and costs associated with each phase of the benchmarking process will efforts to reduce overall benchmarking costs attain their maximum potential payoff.

References

1. Department of the Army, "Development of Standard Benchmarks," Management Information Systems, Information Processing Systems Exchange Pamphlet No. 18-10-2 (May 1973) 1-8.

2. Department of the Navy ADPE Selection Office, "Review of Standard Benchmark Effort," internal memorandum (July 31, 1973).

3. Executive Branch Position Paper, "Proposed Executive Branch Position/Implementation for Recommendation D-14 of the Report of the Commission on Government Procurement" (March 27, 1974).

4. Galler Associates, "An Automated Synthetic Standard Benchmark Technique," Technical Report A-5029, Arlington, Virginia, undated.

5. General Services Administration, "Summary of the NBS/GSA Public Workshop on Remote Terminal Emulation," GSA/ADTS Report CS 76-2 (February 1972).

6. Gosden, J. and R. Sisson, "Standardized Comparisons of Computer Performance," Proceedings of the IFIP Congress (North-Holland Co., 1963) 57-61.

7. Hillegass, J., "Standardized Benchmark Problems Measure Computer Performance," Computers and Automation (January 1966) 16-19.

8. Loring, P., "ADP System Procurement: Concept and Feasibility of an Application Benchmark Library," Mitre Corporation, Technical Report No. 3013 (March 1975).

9. Lucas, H., "Synthetic Program Specifications for Performance Evaluation," Proceedings of the National Conference of the ACM, Vol. 2 (August 1972) 1041-1058.

10. McNeece, J. and R. Sobecki, "Functional Workload Characterization," Proceedings of the 13th Meeting of the Computer Performance Evaluation Users Group, NBS Special Publication 500-18 (September 1977) 13-21.

11. Mitre Corporation, "Approach Plan for a Standard Benchmark Library for Use in Computer System Selection," unpublished report (December 15, 1971).

12. National Bureau of Standards, "Guidelines for Benchmarking ADP Systems in the Competitive Procurement Environment," FIPS PUB 42-1 (May 1977).

13. Oliver, P., et al., "An Experiment in the Use of Synthetic Programs for System Benchmarking," Proceedings of the National Computer Conference (1974) 431-438.

14. Report of the Commission on Government Procurement, Recommendation D-14 (December 1972).

Appendix A

Glossary of Terms

ACCEPTABILITY - A desired combination of qualities of the proposed benchmark library including its proven feasibility (i.e., portability, maintainability, and useability), as defined herein, which would lead ultimately to its use throughout the Federal Government.

APPLICATION PROGRAM - A computer program which directly contributes to the processing of end work, as opposed to computer systems programs, language processors, and other utility programs.

BATCH PROCESSING - A mode of computer processing which is characterized by the concurrent availability to the computer of a complete set of input data for a given job to be processed, the execution of which is not controlled in real-time (i.e., on-line) by a user. See Transaction Processing.

BENCHMARK - A set of computer programs and associated data tailored to represent a particular workload, and used to test the capability of a computer to perform that workload within a predetermined limit.

BUSINESS DATA PROCESSING - A broad class of computer jobs which perform administrative and logistics type functions, and are characterized by heavy demands for data input and output relative to the amount of computation performed. See Scientific Computing.

EVALUATION CRITERIA - The set of measurement standards (to be) established as a part of this study as a basis for evaluating the degree to which proposed solutions satisfy real or potential technical deficiencies of a benchmark library.

FEASIBILITY (of a benchmark library) - The technological capability to establish and maintain a usable set of synthetic benchmark programs that can be assembled and adjusted to represent large classes of Federal computer

workloads. See Usable.

FUNCTIONALLY-DESCRIBABLE WORKLOAD - A computer workload which can be characterized and quantified in terms of well-defined and predictable processing functions. See Resource-Oriented Workload.

LIBRARY (benchmark library) - A collection of synthetic benchmark programs which have been tested and documented for general use by Government agencies in computer benchmarks. See Synthetic Benchmark Program.

MAINTAINABLE - The requirement that a benchmark library be supported by systems to test and document additional library programs, to respond to deficiencies, and to update the programs as a result of changing technology.

MIX - A combination of different benchmark programs and data which together correctly represent the real workload.

PORTABLE - A requirement of synthetic programs in the benchmark library to represent a specified amount of work on different computers without undue bias resulting from differences among the computers and their systems software. Also refers to the ability of benchmark programs to run on different systems with little or no source-code changes.

QUANTIFY - With respect to a computer workload, the process of expressing the workload in numerical values.

REPRESENT - The ability of a benchmark to impose the same demands on a computer system as the real jobs which will be processed on that system during a given time frame.

RESOURCE-ORIENTED WORKLOAD - A computer workload which is characterized and quantified in terms of its consumption of computer resources. See Functionally-Describable Workload.

SCIENTIFIC COMPUTING - A broad class of computer jobs which involve extensive mathematical functions and are

characterized by heavy demands for computation relative to the amount of data input and output performed. See Business Data Processing.

SYNTHETIC BENCHMARK PROGRAM - A parameterized, functional computer program designed to represent a particular class or function of application programs for benchmarking purposes only; the synthetic benchmark program serves no other useful function.

TRANSACTION PROCESSING - A mode of computer processing in which data is available as a function of time, usually when the transactions result from an on-line user. See Batch.

USABLE - The ability of the potential library of synthetic benchmark programs to represent an applicable computer workload. A necessary ingredient is an effective method of analyzing and mapping the workload quantification to units which are compatible with the synthetic program parameters.

Appendix B

USDA Workload Mapping Procedure

Preface

The following material has been extracted from the USDA benchmark system documentation. It is not presented here as a stand-alone procedure, since the complete documentation and some tutoring would be required to follow the procedure.

1. Derive Benchmark Workload

The benchmark workload is somewhat unique in its objective to establish the processing capacity of the system. That is a different objective than cost justification, i.e., calculating the value of the system, which is concerned with all work which the computer will do. The benchmark will be based upon the projected workloads during periods of maximum throughput, which tend to recur in daily, weekly, monthly, or annual patterns. The activities described below are necessary to quantify this workload.

(a) Identify quantifiable events which represent agency functions. These functions must be major agency program or administrative functions. The proper level of detail for these functions is the highest one which can result in an explicitly determinable set of ADP operations. A Commodity Credit Corporation loan, for example, is not sufficient detail, because there are many kinds of such loans, requiring different processing. The output of this activity will be a list, for each agency, of the agency workload functions, and the specific events to be quantified for each, i.e., applications processed or loans made.

(b) Identify and define benchmark ADP operations. A benchmark ADP operation will be directly and explicitly represented in the benchmark workload mix by a synthetic program or some other workload category. Not all programs in the library have to be included, and there are some workload categories which cannot be represented by synthetic programs. For example, there may be high volume ADP applications which are too complex to represent in synthetic programs. Other categories of work, such as compiling, sorting, and data base query language operations, will use vendor software exclusively. The output of this activity will be a list, with descriptions, of the ADP operations

likely to constitute significant parts of the peak workloads to be benchmarked. A single composite list will apply to all agencies. It is possible that one or more of these operations might prove to be insignificant when the peak periods are finally identified and quantified, and might then be omitted from the benchmark.

(c) The volume for each agency quantifiable event identified in activity 1 (a) must be projected over the scheduled life of the computer system. Quantification for each year is required for each item. More detailed quantification is also necessary for workload items which experience cyclical ups and downs of volume within a year. If the same cycle is repeated annually, a single profile giving the workload percentage occurring in each month will suffice for all years. Still shorter cycles may be expected, in particular, daily cycles for on-line workload. A single profile of daily clientele arrival rates may be provided for all those on-line functions triggered by the public at distributed locations. The output of this activity will be, first of all, a columnar chart with agency quantifiable events (by code and name) down the left side and workload across the top, as shown in the Workload Projection Form, Figure 1. Second, more detail will be provided, by hour of day, or other period, to show volume cycles of shorter duration. The two kinds of projections will make it possible to project workload for any particular point within the scheduled system life.

(d) Determine, by analytical means, the relationships between quantifiable events specified in activity 1 (a) above, and the benchmark ADP operations identified in activity 1 (b). These relationships must be mapped into a matrix which lists the ADP operations on one axis and quantifiable events on the other, as shown on the Workload Mapping Form, Figure 2. Experience indicates that ADP systems which support agency functions fall into three categories for mapping, defined and treated as follows:

(1) There is a category of ADP systems which are executed frequently, at least monthly, and workload is a direct function of the quantifiable events. ADP systems must be further divided into contiguous subsystems; that is, where processing by a single subsystem is performed without intervening gaps in time. Identify as category 1, and list, for each

USDA QUANTIFIABLE BENCHMARK EVENTS
AND WORKLOAD PROJECTIONS

Agency _____ Date _____

Quantifiable Event		Volume Per Year						Percent Per Month											
Code	Name	1977	1978	1979	1980	1981	1982	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC

Figure 1. Workload Projection Form

USDA ASSIGNMENT OF BENCHMARK TO ADP OPERATIONS Agency ______ Date ______

Code for Quantifiable Event	ADP System or Subsystem	ADP Representative and Phone Number	Category	Disp/Freq.	1. ORSIND	2. ORNDEX	3. OUSIND	4. OUNDEX	5. OENTRY	6. OCOMPU	7. BRSIND	8. BUSIND	9. BUNDEX	10. BUMULT	11. BRSERL	12. BUSERL	13. BRSEQU	14. BSEDIT	15. BRNDEX	16. BCOMPU	17. SORT Merge	18. DBL Queries	19. Report Gen.	20. COBOL	21. FORTRAN	22. IPD

Figure 2 Workload Mapping Form

subsystem:

- o Code assigned to quantifiable event from list 1 (a) above.
- o ADP system/subsystem name.
- o Name and phone number of ADP consultant.
- o Category (i.e., "1") of system/subsystem.
- o Displacement (time) in months from incidence of event to processing.
- o Under each benchmark ADP operation, the number of executions per incidence of event, for the entire life of the transaction.

(2) The second category of systems/subsystems is those for which there is infrequent (quarterly, semi-annual, or annual) ADP processing, and workload is a direct function of quantifiable events. ADP support systems must be further divided into subsystems by processing frequency. Specify the same as category 1 above except identify as category 2, and use one of the following frequency codes in lieu of displacement:

Code	Frequency
Ø	Processed annually at end of calendar year
1	Processed annually at end of fiscal year
2	Processed semi-annually
4	Processed quarterly

(3) The final category consists of systems/subsystems for which workload is not a function of a quantifiable event. Maximum flexibility is provided for quantifying and mapping this workload, using a combination of the Workload Projection and Workload Mapping forms. Show category 3 for these systems/subsystems. The displacement frequency column is not used in tallying the workload and may be used as desired for its information content. The distribution of workload will be derived from monthly percentages provided on the Workload Projection Form. The best way to learn how to quantify and map category 3 workload is to understand how it will be tallied. For a given month, the monthly percentage will be multiplied by the appropriate annual workload projection. This product will in turn be multiplied by each of the ADP operation quantities for designated systems/subsystems to yield workload for the month in question. Given the three

value fields to be multiplied together, the actual quantities can be manipulated in a variety of ways to produce the same results. As with category 1 and 2 system/subsystems, category 3 line items on Workload Mapping Forms are associated with workload projections by using the same quantifiable event code.

(e) Select peak workload months. The objective of this activity is to identify the peak months of computer workload for the combined agencies. This will be done by tallying workload for each month from Workload Projection and Mapping forms. Detailed methodology cannot be worked out in advance because the complexity of the task depends upon all the data collected in activities 1 (a) through 1 (d). If all ADP operations peak at the same time, then the selection will be obvious. More analysis will be required if disparate peaks materialize. Management guidance must be obtained as to the desired level of capability to support peak periods, in order to determine how much flattening of peaks is appropriate. The output of this activity will be the identification of at least two representative peaks, occurring in the first and final years. If the workload changes between these years in volume or composition, in other than approximately linear fashion, additional peaks must be identified to represent the changes.

(f) Quantify peak periods. Using the data derived in steps 1 (c) and 1 (d), calculate the aggregate number of iterations of each benchmark operation, for all combined agencies, required to perform the agency workload during each of the peak periods. The output of this activity will be a quantification table for each peak period, giving the number of iterations for each of the benchmark ADP operations.

(g) Determine benchmark transaction characteristics. For the purpose of this discussion, a transaction is a coded representation of an event which triggers one iteration of one of the benchmark ADP operations discussed in paragraph 1 (b) above. This definition will apply whether the ADP operation is on-line or batch, the difference being whether the transactions are presented to the system individually at the times when the driving events occur, or collected into batches for processing. This activity will require determining the characteristics of the transactions likely to be in the operational systems and assuring that these

characteristics are adequately represented in the benchmark programs.

(h) Determine data storage needs and characteristics of the data base. This activity will consist of determining the size of the data base(s) to be stored in the object computer system, and the characteristics of the major data files. It will also require taking measures to assure that the benchmark adequately represents these data characteristics.

2. Analyze Workload

The purpose of this analysis is to translate the workload projections into parameters for the benchmark. These specific activities will be required:

(a) Derive synthetic program parameters. These include the sizes of programs, rate of job execution, numbers of statements executed in each program, number of copies of each program, and transaction rate per copy.

(b) Develop data storage benchmark plan. The size of the data base, number and sizes of files, and file organizations must be decided.

(c) Associate programs, transactions, and data files. Decide the ratio of matching data base records to transactions for each transaction type.

(d) Derive data generation parameters. Attempt to assign keys which will render the correct transaction-to-data-base ratio, and at the same time yield the proper data volumes.

3. Develop and Test Benchmark Materials

This is a group of activities extending over the total duration of the benchmark effort, related in that they require knowledge of the benchmark programs and use of computers. Specific activities are:

(a) Construction of emulators. In order to test synthetic programs on USDA computers, a set of software emulators is required to perform the functions of the transaction processor and data base management system. This activity consists of constructing and/or modifying these emulators for the current procurement and testing them.

(b) Retest all benchmark components. This activity consists of generation of test transaction and data via the data generator and exercizing all emulators, synthetic application programs, and the post processor.

(c) Update synthetic programs in accordance with new specifications.

(d) Modify data generator to produce transactions and data files in accordance with new specifications.

(e) Generate new transactions and data.

(f) Test benchmark and produce control values.

(g) Reproduce materials for vendors. Use a tape copy process. Use each new copy to reproduce the next, finally validating the last copy against the original.

Tally Process

A computerized process will tally the workload for any given month in the scheduled system life, from data provided on Workload Projection and Workload Mapping forms. The results will be an aggregate volume for each ADP operation listed on the mapping forms. Detailed steps for the tally, with a year and month given as parameters, are:

1. Initialize a tally for each of the benchmark ADP operations.

2. Process each agency quantifiable event sequentially through steps 3 and 4.

3. Get the workload projection for that event and hold.

4. Process each ADP subsystem for the function according to which of the three categories it is in, e.g.,

(a) For category 1, subtract displacement (see 1.d.1) from parameter to obtain month of workload origin. If it falls earlier than available data, add 12 months. Obtain quantification projection for month of origin. Multiply the number of executions of each ADP operation by the quantification for the month of origin and add to their respective tallies.

(b) The second category is periodic processing with frequency codes of 0, 1, 2, or 4. The workload for a given system/subsystem will be used only if part of the processing is scheduled to fall in the month for which the tally is being made. That can be determined from Table 1, which shows an example of the allocation for each frequency code to months. If the allocation is non-zero for the object month, then the combined workload for the months listed in the corresponding "use data for" column of Table 1 is determined. That is done by multiplying each month percentage by the appropriate annual volume and summing the products. The allocation for the object month is then applied to the sum. This product is then multiplied by the number of executions of each ADP operation, and the products are added to their respective tallies.

(c) Category 3 line items are treated as those in category 1, except that displacement is assumed to be 0. See paragraph 1.d.3 for a discussion of the use of

DISTRIBUTION OF PERIODIC WORKLOAD

Category 2

Allocate to:	Frequency 0 Use Data for:	Allocation	Frequency 1 Use Data for:	Allocation	Frequency 2 Use Data for:	Allocation	Frequency 4 Use Data for:	Allocation
Dec	Jan-Dec:	10%			July-Dec:	5%	Oct-Dec:	5%
Jan		50%				35%		80%
Feb		40%				10%		15%
Mar					Oct-Mar:	5%	Jan-Mar:	5%
Apr						35%		80%
May						10%		15%
June					Jan-June:	5%	Apr-June:	5%
July						35%		80%
Aug						10%		15%
Sept			Oct-Sept:	10%	Apr-Sept:	5%	July-Sept:	5%
Oct				50%		35%		80%
Nov				40%		10%		15%

Table 1. Distribution of Periodic Workload (Category 2)

category 3.

5. Print out the final tallies for each ADP operation.

Appendix C

Sample Evaluation Criteria

The following describes a proposed set of evaluation criteria to be used to determine the useability and portability of a candidate benchmark library.

1. Useability

1.1. Background

Recall the definition of "usable" (see Appendix A):

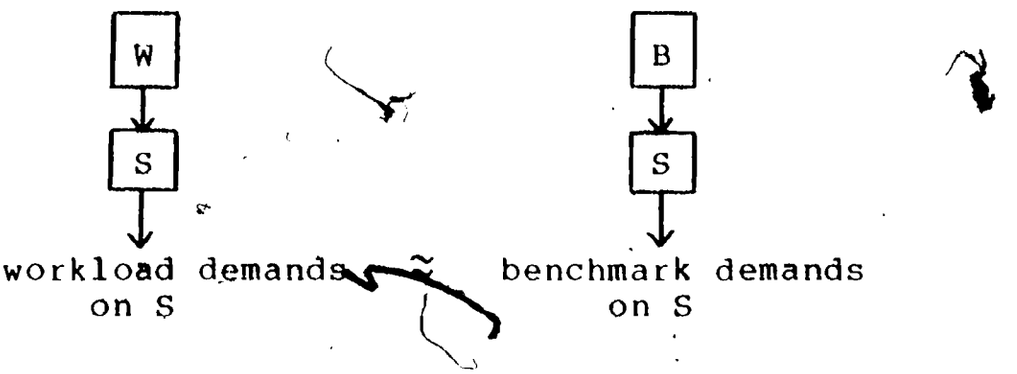
> USABLE - The ability of the potential library of synthetic benchmark programs to represent an applicable computer workload. A necessary ingredient is an effective method of analyzing and mapping the workload quantification to units which are compatible with the synthetic program parameters.

Implicit in this definition are two necessary components of the library: (1) a set of programs that can represent a workload; and (2) a set of procedures that specify how to use the library. Thus, any evaluation criteria testing "useability" should test both of these capabilities.

Recall also the definition of "represent":

> REPRESENT - The ability of a benchmark to impose the same demands on a computer system as the real jobs which will be processed on that system during a given time frame.

This requirement is summarized by the following diagram:



That is, for any given system S, if the workload W and the benchmark B are run on S, then W and B should produce approximately the same demands on S.

The next question is, what do we mean by "the same demands." The following three requirements define what it means for "W and B to produce approximately the same demands on S":

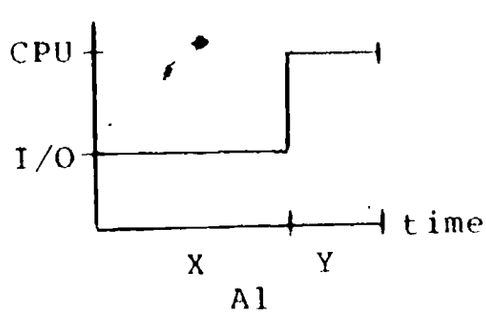
1. The elapsed running time of W on S should be approximately the same (e.g., within 10%) of the elapsed running time of B on S. Note, for on-line applications, "elapsed time" could be replaced by "response time."

2. The resource utilization data (e.g., percent CPU active, average disk space used, I/O volume transferred) when W is run on S should be approximately the same as the corresponding data when B is run on S.

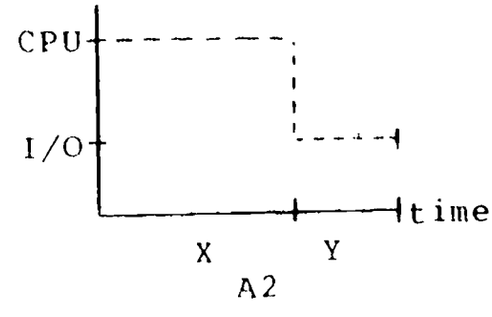
3. The resource profiles of W and B should be approximately the same.

Items 1 and 2 seem obvious if one wants B to properly size the right system. It is item 3, however, which requires expanded discussion. In order to show the importance of item 3, especially in a multi-programming environment, assume the following situation:

1. Let two applications, A1 and A2, make up the real workload W and have the following resource profiles:



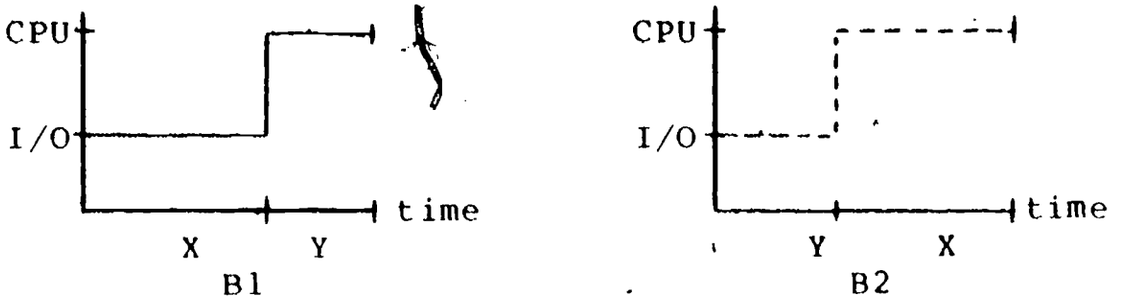
A1



A2

That is, A1 uses X seconds of I/O followed by Y seconds of CPU, and A2 uses X seconds of CPU followed by Y seconds of I/O.

2. Assume that benchmarks B1 and B2, which are claimed to represent A1 and A2 respectively, have the following resource profiles:

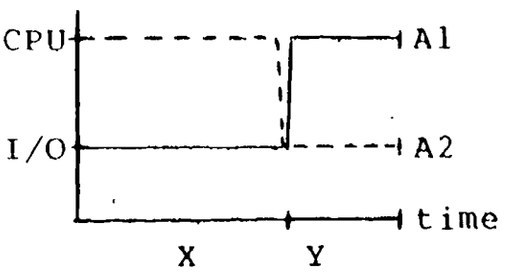


Note that:

(a) both B1 and B2 have the same elapsed times as the applications they each claim to represent; and

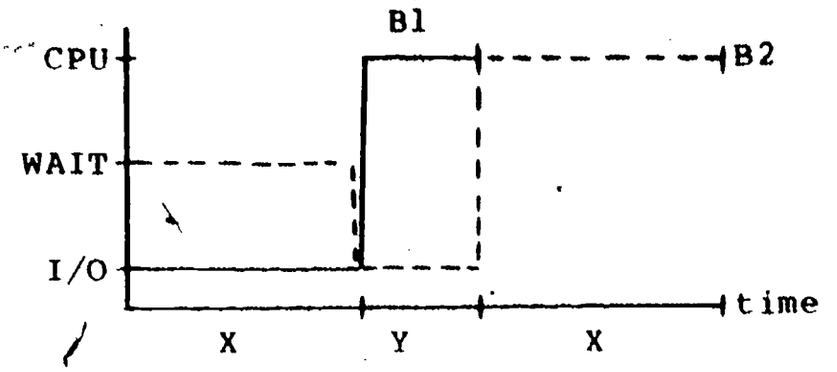
(b) both B1 and B2 have the same resource utilization data (i.e., CPU and I/O times) as the applications they claim to represent. In addition, note that B1 has the same profile as A1, but B2 and A2 have different profiles.

3. Assume both applications are now run in a multi-programming environment where the CPU and I/O can overlap each other:



Note that the total workload completes in elapsed time: X+Y.

4. Assume both benchmarks B1 and B2, which claim to represent A1 and A2, are now run in the same multi-programming environment:



Because B2 had to WAIT for B1's I/O demands to complete, the elapsed time to run the total benchmark was extended to: 2X+Y -- nearly double that of the workload which the benchmark claimed to represent.

The above example thus points out that it is not sufficient for a benchmark to have the same elapsed time and resource utilization data as the workload it claims to represent; but rather, the benchmark should also have a resource profile similar to that of the real workload -- especially in a multi-programming environment.

1.2. Useability Evaluation Criteria

Based on the previous discussion, the following evaluation criteria would thus determine whether a candidate benchmark library is acceptable in terms of "Useability":

Useability Criteria: A benchmark library is "usable" if, given an arbitrary workload W, programs from the library can be selected, configured (i.e., parameters properly set), and combined in such a way, using established library procedures, so that the collection of programs (i.e., the benchmark B) suitably represents W. That is, for any arbitrary system S:

a) the elapsed time of W on S $\approx$ the elapsed time of B on S;

b) the resource utilization of W on S $\approx$ the resource utilization of B on S;

c) the resource profile of W on S ≈ the resource profile of B on S.

1.3. Application of Useability Evaluation Criteria

Having defined the evaluation criteria which will determine whether a candidate benchmark library is usable, the next step is to define the procedure for applying the criteria. This section will outline a sequence of steps to be followed which will determine whether a candidate library meets the Useability Criteria for a given workload on a given system. Note, the ideal test of a library would be to apply the Useability Criteria across all workloads and across all systems. Because this would not be practical, the procedure actually defines a set of necessary, but not sufficient, conditions for useability.

Before the procedure which will determine useability can be applied, the following preliminary steps should be performed in order to obtain a test workload W:

1. Identify ADP functions {F1,....,Fn} common to many agencies, by:

   (a) surveying agencies - e.g., distribute a list of ADP functions (e.g., those identified by Lucas [9]) and have agencies indicate the frequency of use and importance of each;

   (b) or, alternatively, identify those functions believed to be used by many agencies and see if this list is consistent with recent RFP's.

2. Select from an agency (or create) a set of applications {A1,...,An} that perform the functions identified in 1. These applications are real computer programs that will make up the test workload W. Note, each Ai could be composed of many programs.

Having constructed a test workload W, the following steps are performed to determine the "useability" of a candidate benchmark library. The following procedure is optimal in the sense that if a benchmark library will fail, it will fail early.

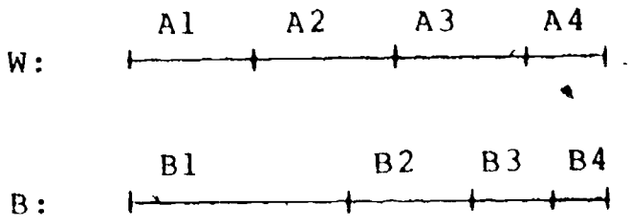
Procedure to Determine Useability

1. Using the benchmark library procedures, create a benchmark Bi (a set of library programs) to represent each application Ai of W. That is, apply the library procedures to choose the proper programs and parameter settings. Call the collection of Bi's, B.

2. Run W and B single thread (i.e., not multi-programmed) on several large systems and calculate the errors in demands as follows:

A. Elapsed Times

a) Determine the elapsed times of each Bi and its corresponding Ai. Note, it is necessary to look at individual (Bi, Ai) differences and not just total (B, W) elapsed time differences since errors could have a cancelling effect, as illustrated in the following elapsed time charts:



Here, cumulative elapsed times for W and B are the same, but individual ones are not.

b) For each system on which W and B are run, calculate the maximum elapsed time relative error:

$$\text{System 1:}\quad E1 = \max\left(\frac{|A1-B1|}{A1}\ ,\ \frac{|A2-B2|}{A2}\ ,\ \ldots\right)$$

$$\text{System 2:} \quad E2 = \max\left(\frac{|A1-B1|}{A1}, \frac{|A2-B2|}{A2}, \ldots\right)$$

c) Find the maximum elapsed time error across all systems:

$$E = \max (E1, E2, \ldots)$$

Thus, E represents the maximum percent error that ever occurred between an application and its corresponding benchmark. For example, if the following represented elapsed running times in minutes:

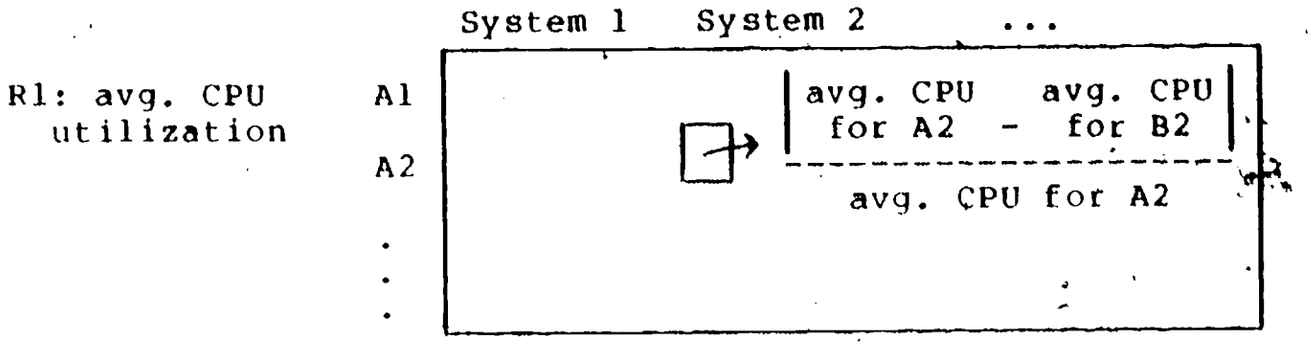
	A1	B1	$\frac{\lvert A1-B1 \rvert}{A1}$	A2	B2	$\frac{\lvert A2-B2 \rvert}{A2}$
System 1	10	15	.50	12	9	.25
System 2	14	15	.07	11	12	.09
System 3	13	13	.0	9	8	.11

then E would equal .50, i.e., the maximum relative error across all systems and (application, benchmark) pairs.

B. Resource Utilization Data

a) For each major system resource Ri (e.g., R1 = CPU, R2 = core, R3 = disk space used, R4 = channel activity, ...), collect appropriate utilization data when W and B are run on each system.

b) For each resource, calculate the resource utilization errors between each application and its corresponding benchmark; for example,

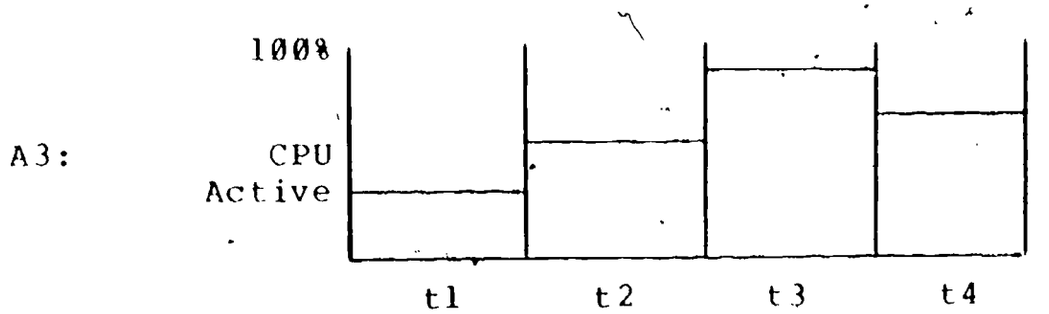


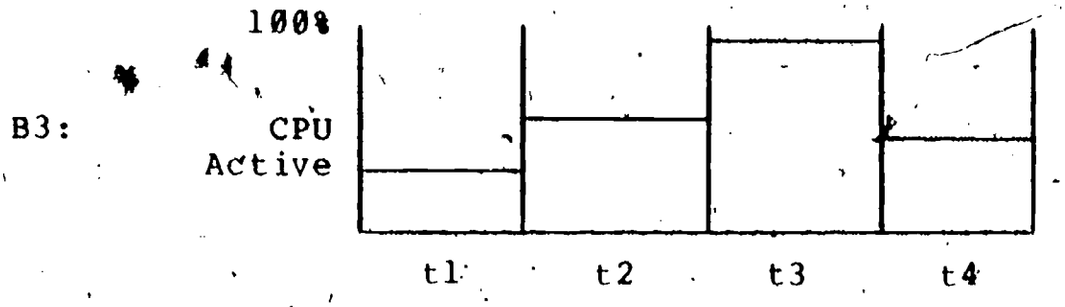
c) Find the maximum resource errors across all systems. Construct a resource utilization error vector:

R = (max. CPU utilization error,
max. core utilization error,...)

C. Resource Profile Data

a) For each major system resource, obtain a profile across time of resource usage for each application and its corresponding benchmark. For example, on System 2 the CPU profiles for A3 and B3 might look like:





b) Apply statistical techniques to all profiles for each resource and determine the profile pairs least like each other. Quantify this discrepancy in terms of relative error or confidence limits.

c) Construct a profile error vector:

P = (max. CPU profile error,
max. core profile error,...)

In summary, the value E and the vectors R and P thus tell, in quantifiable terms, how close (in demands) B is to W.

3. Determine if B has passed the usebility test thus far. That is, see if E, R and P are within acceptable bounds (e.g., E $\leq$ 10%). If not, the candidate benchmark library fails. If B passes so far, continue with the next steps.

4. Construct a transaction processing test workload W. Repeat steps 1-3. If pass, continue below.

5. Try a combination batch and transaction processing workload and repeat steps 1-3. If pass, continue.

6. Try all of the above in a multi-programming environment.

The above procedure will determine if a candidate benchmark

library can adequately represent existing application programs. A further question is how close the benchmark library can come to representing applications specified with less and less knowledge -- i.e., closer to the functional specification level.

2. Portability

2.1. Background

Recall the definition of "portable":

> PORTABLE - A requirement of synthetic programs in the benchmark library to represent a specified amount of work on different computers without undue bias resulting from differences among the computers and their systems software. Also refers to the ability of benchmark programs to run on different systems with little or no source-code changes.

Thus, the benchmarks constructed from the library must have two necessary qualities: (1) they must contain standard language and data constructs; and (2) they must not "unduly bias" one system over another. It is clear what the first criterion means. What is not clear is the meaning of "unduly bias." The following discussion addresses this latter point.

A benchmark should adequately represent a workload so that the ability of one system to handle the workload better than another system is reflected in the benchmark running times, resource usage, etc. That is, the benchmark should reflect the same "natural biasing" that will take place when the real workload is run -- this, after all, is what benchmarking is all about. The problem, of course, is that the benchmark should not perform additional activities which are not needed to represent the workload since these additional activities are subject to different system transformations and hence may skew the benchmark results.

How does one then determine if a benchmark is performing these "additional activities" -- that is, if it is unduly biased? One of the only practical ways is to

determine if the benchmark is placing more resource demands on the system than the real workload would. The assumption here, of course, is that if the benchmark were performing "additional activities" they would be reflected in additional demands. This assumption appears correct except in the case in which additional (or insufficient) demands cancel each other with the net effect that the benchmark has similar aggregate demands as the workload, though different activities. Furthermore, it is necessary to assume that the application programs from which the benchmarks are constructed are themselves unbiased.

2.2. Portability Evaluation Criteria

The following evaluation criteria would thus determine whether a candidate benchmark library is acceptable in terms of "portability":

> Portability Criteria: A benchmark library is "portable" if, given an arbitrary workload W, a benchmark B can be constructed which:
>
> a) contains only standard language and data constructs; and
>
> b) does not place additional demands on an arbitrary system S as would W if W were actually run on S (i.e., does not unduly bias one system over another).

2.3. Application of Portability Evaluation Criteria

The procedure for applying the Portability Criteria to a candidate benchmark library can, as it turns out, be performed in parallel with the "useability" test described earlier. Having constructed a benchmark B to represent a test workload W, B could be examined either manually or automatically to determine if it contains any non-standard language constructs. Secondly, during the running of W and B on various systems, the resource utilization and profile data collected for the "useability" tests can also be used to determine whether (and how much) B is unduly biased (since, as has already been stated, unduly biased means B places different demands on the system than does W). In fact, the resource utilization error matrix developed earlier will tell whether the benchmark is biased by application (comparing the matrix rows) or by system (comparing the matrix columns).