NBS TECHNICAL NOTE **920**

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

Use of Synthetic Benchmarks for Estimating Service Bureau Processing Charges

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards [1] was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau consists of the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Institute for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of the Office of Measurement Services, the Office of Radiation Measurement and the following Center and divisions:

Applied Mathematics — Electricity — Mechanics — Heat — Optical Physics — Center for Radiation Research: Nuclear Sciences; Applied Radiation — Laboratory Astrophysics [2] — Cryogenics [2] — Electromagnetics [2] — Time and Frequency [2].

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement, standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; and develops, produces, and distributes standard reference materials. The Institute consists of the Office of Standard Reference Materials, the Office of Air and Water Measurement, and the following divisions:

Analytical Chemistry — Polymers — Metallurgy — Inorganic Materials — Reactor Radiation — Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations leading to the development of technological standards (including mandatory safety standards), codes and methods of test; and provides technical advice and services to Government agencies upon request. The Institute consists of the following divisions and Centers:

Standards Application and Analysis — Electronic Technology — Center for Consumer Product Technology: Product Systems Analysis; Product Engineering — Center for Building Technology: Structures, Materials, and Life Safety; Building Environment; Technical Evaluation and Application — Center for Fire Research: Fire Science; Fire Safety Engineering.

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in improving cost effectiveness in the conduct of their programs through the selection, acquisition, and effective utilization of automatic data processing equipment; and serves as the principal focus within the executive branch for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Institute consists of the following divisions:

Computer Services — Systems and Software — Computer Systems Engineering — Information Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal Government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System; provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data — Office of Information Activities — Office of Technical Publications — Library — Office of International Relations — Office of International Standards.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
[2] Located at Boulder, Colorado 80302.

Use of Synthetic Benchmarks for Estimating Service Bureau Processing Charges
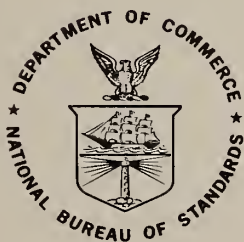
t Technical Note no. 920

Dennis M. Conti

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

TABLE OF CONTENTS

USE OF SYNTHETIC BENCHMARKS FOR ESTIMATING
SERVICE BUREAU PROCESSING CHARGES

by

Dennis M. Conti

ABSTRACT

This report describes the development of a new synthetic
benchmark technique for estimating batch processing charges
at service bureau sites. This technique was used to esti-
mate the cost of processing a large batch workload at a
number of service bureaus within the same mainframe family.
The method was found to be low-cost, yet reasonably accurate
for a certain class of service bureau charging algorithms.
Refinements of this method are suggested which will extend
its applicability to other algorithms. The procedures used
to create and run the benchmark, together with the projection
of total workload cost are described.

Key Words: Benchmarking; charging algorithms; service
            bureaus; synthetic benchmarking; workload
            characterization.

I. INTRODUCTION

Federal agencies with a requirement for ADP products and services
and subject to Office of Management and Budget Circular A-76, "Policies
for Acquiring Commercial or Industrial Products and Services for
Government Use,"[1], may find it necessary to compare the cost of data
processing at commercial service bureaus and the cost of doing the same
processing in-house at Federally-owned and operated facilities. This,
in turn, has created a requirement for a low-cost, reasonably accurate
method which agencies can use to project their processing costs at
various service bureau sites. This report describes preliminary work
undertaken by the National Bureau of Standards to develop such a
method.

For input to an A-76 cost study it was undertaking, the Data
Management Center (DMC) of the Department of Health, Education, and

1

Welfare requested that the Institute for Computer Sciences and
Technology of the National Bureau of Standards (NBS/ICST) assist it in
estimating the cost to process its batch workload at commercial service
bureau sites. The scope of the project was to include only direct batch
processing charges--no attempt would be made to estimate teleprocessing
or other ancillary charges (e.g., storage charges for tapes, terminal
connect charges, etc.). Furthermore, the study was to consider only
service bureaus with DMC-compatible systems (IBM OS and VS systems in
the 370/155-370/168 range). Two approaches were investigated, as
described below.

The charging algorithms of a number of service bureaus were to be
obtained by NBS/ICST and coded into a COBOL program. The system
accounting log data (SMF [2]) for a representative month was to be
processed by this program and an estimate of service bureau charges,
based on their respective algorithms, was to be obtained. This method
did in fact produce very accurate, job-by-job cost projections for each
service bureau [3]. Two difficulties with this method, however, were:
(1) the need to keep the highly-sensitive service bureau algorithms
confidential; and (2) the difficulty in verifying that the COBOL version
of the algorithms, and indeed the version supplied by the service
bureaus, were valid.

The second approach was to develop a method by which the DMC's
batch workload could be represented by a small set of synthetic benchmark
programs. The programs would be run at the same set of service bureaus
selected for the algorithm approach. The total DMC workload charges
would then be projected from the actual charges incurred by the bench-
mark jobs. The purpose of the synthetic benchmark approach was not only
to provide a validity check against the algorithm approach, but also to
determine if a low-cost, yet reasonably accurate method for estimating
batch processing charges could be developed.

The objective of this report is to describe the synthetic benchmark
approach taken to estimate the cost of processing DMC's workload. Other
agencies may find these techniques, or extensions thereof, useful in
their own procurement environment.

This report begins with a description of previous attempts to model batch workloads with synthetic programs (primarily for the purpose of thruput estimation), and the relationship of this past work to the approach taken here. Two initial attempts to construct an acceptable synthetic mix are then described. A detailed discussion then follows of the assumptions and theory underlying the approach ultimately taken. Finally, the procedures used to run the synthetic programs and to project the total workload processing costs are described.

II. BACKGROUND

Synthetic benchmarks can be classified into two categories: task-oriented and resource-oriented. Task-oriented synthetics are artificial programs which attempt to model the functions of the real workload, while resource-oriented synthetics attempt to model the resource demands which the real workload places on a system. Because of the difficulty in obtaining functional descriptions for most jobs in a workload, resource-oriented synthetics have been investigated more extensively.

Historically, resource-oriented synthetics have been used to tune existing systems or to benchmark new systems. In 1969 Buchholz [4] described a parameter-driven PL/I program which he claimed could be used as "a well-behaved exerciser of system features or a tool for comparing the speed of dissimilar systems." The Buchholz program was a generalized file maintenance program which operated on two ordered files, a master file and a detail file. The program would sequentially read the master file until a record was found which matched the current detail record. Upon detection of a match, a compute-bound kernel would be executed. An attempt would then be made to match the next detail record. This process continued until the end of the master file was reached. Input parameters could be chosen to control the number of master and detail records, and the number of times the compute kernel was executed following a master-detail match. The Buchholz program has been used as the basis for many resource-oriented synthetic benchmarks.

Crowding [5] attempted to represent a real workload by: (1) categorizing user jobs into core classes; (2) computing the "average job" for each class; and (3) constructing a resource-oriented synthetic job to represent each class. These synthetic jobs were then combined to represent the total workload. Crowding was able to obtain close agreement between machine utilization data for the real workload and that for the synthetic job stream on a single system. Kernighan and Hamilton [6] also achieved success when they attempted to represent a workload by a set of synthetic jobs which exercised only the CPU, IO, and core. Furthermore, they felt that "moving [their] benchmarks from one machine or system to another is simple" and that their approach was "quite amenable to comparisons of system 1 on machine A with system 2 on machine B."

One of the latest and more sophisticated uses of synthetic benchmarks is described by Sreenivasan and Kleinman [7]. They attempted to represent the real workload by a joint probability matrix of CPU time and IO counts. That is, for each (CPU, IO) pair the probability was calculated that a job existed in the real workload with processing time 'CPU' and EXCP[1] count 'IO'. A "drive workload" consisting of a Fortran version of the Buchholz synthetic was constructed to match this CPU-IO distribution. Since most of the jobs in the real workload required a 128K byte partition, all of the synthetic jobs were forced to run in 128K bytes. Sreenivasan and Esposito [8] later extended this work in a study of a Burroughs system to include core as a third variable to be modeled. They concluded that using their approach, "the relative performance of dissimilar systems can be evaluated." King [9] describes an automated procedure which models the real workload by matching the resource requirements of a set of synthetic jobs with the resource requirements of user jobs in the real workload. He concludes that this automated procedure solves, among other things, the "transportation problem" of benchmark jobs to a "strange environment."

These previous attempts to model a real workload by synthetic jobs had as their primary goal the evaluation of system performance.

---

[1]EXCP - a measure of the number of IO block transfers on IBM systems.

Furthermore, several of these authors have suggested that resource-oriented synthetics can be used in the comparison of dissimilar systems. The use of resource-oriented synthetics for this purpose has been one of the most controversial issues of synthetic benchmarking which others have questioned [10,11]. While the approach used in this study employed resource-oriented synthetics to model the DMC workload, its use was restricted to estimating processing costs within a single family of systems. These were two distinct differences from the previous approaches which used synthetics to place a representative load on a system (or on systems across vendors) in order to obtain thruput and turnaround times. In the study described here, resource-oriented synthetics are used only to drive charging algorithms in a representative manner.

III. PRELIMINARY EFFORTS

The first two attempts to construct a synthetic benchmark representative of DMC's workload proved unsatisfactory for a number of reasons. Either the cost to run the benchmark at each service bureau would have been excessive, or else the benchmark would not accurately represent the DMC workload. However, a detailed analysis of the general nature of the service bureau algorithms provided the insight that allowed for the construction of a reasonably accurate, low-cost synthetic benchmark. This section first describes the required SMF data preparation steps and then discusses the initial attempts to produce an acceptable synthetic mix.

A. SMF Data Preparation

A major assumption of this study was that the workload characteristics captured by SMF were typical of the DMC workload and could serve as the primary basis for dollar charges at commercial service bureaus. Before application of either the algorithm or synthetic approach could commence, therefore, a data base consisting of DMC's SMF records for the month of July 1974 (chosen by DMC as a representative month), grouped by job, was

5

required. In addition, certain anomalous cases (e.g., step records with no corresponding job termination record) needed to be identified, and all improper records eliminated. The following describes the sequence of steps taken to construct this final data base.

SMF tapes were provided at the outset of the study which contained all SMF record-types collected at DMC during the month of July 1974. In addition, all account numbers and job names were suitably encoded. Because they were the only records of interest in this study, the following two record-types were extracted from these tapes:

Type 4 - step termination record - 89,700

Type 5 - job termination record - 26,352

See the IBM SMF manual [2] for a complete description of each record type.

The SMFCOPY program (a utility provided by DMC) was used to extract the above records for batch jobs only, thereby reducing the number of records to the following:

Type 4 - 89,084

Type 5 - 25,850

This intermediate file was sorted first by job-log-number (i.e., job name, reader start time, reader start date), and then by record time-stamp within each job. Because of the volume of data, two sorts and then a merge were actually required.

Finally, this set of data was examined for consistency (every type 4 had a corresponding type 5, every type 5 had at least one 4), and extraneous records (type 4 primarily) were deleted. This resulted in a final data base containing 86,392 step termination and 25,850 job termination records.

B. Initial Attempts

Before describing the synthetic approach actually used in this study, two attempts to produce a representative, yet low-cost synthetic mix will now be discussed.

The first such attempt consisted of the following steps:

1. The type 4 records were partitioned into the three highest

revenue-producing job classes: H, D, and J. A fourth class, OTHER, included the remaining job-step records.

2. For each job class, the job-step records were divided into core groups of 40K increments.

3. For each core group, a CPU-IO joint probability matrix was computed.

4. A synthetic mix of N jobs was then constructed whose joint probability matrix matched that of the real workload, but whose total CPU time did not exceed one hour.

5. This mix of N jobs was then to be run at the selected service bureaus. Appropriate expansion factors were to be applied to the individual job costs to determine total projected workload costs.

The above approach closely paralleled the synthetic benchmark generation process outlined by Sreenivasan and Kleinman [7]. Several difficulties were encountered with this approach, however. First, only the dominant (CPU, IO) pairs could be represented in the final synthetic mix because of the need to satisfy the one hour total CPU time criterion. Second, even with this final mix, minimum job charges at several service bureau sites would have overly biased the resulting projected workload costs. A modified version of this approach was then tried.

Each (CPU, IO, core) combination was first ranked according to total resource usage (i.e., the sum of CPU time, EXCP counts, and allocated core). The relative contribution of each of the four job classes to total workload CPU time was calculated. The proportionate CPU contribution of each class to a total mix time of one hour was then computed. Figure 1 depicts these results.

Class	Real Workload CPU Time (sec)	% of Total	Synthetic Mix Allotted CPU Time (sec)
H	192,114	31.2	562
D	71,338	11.6	209
J	100,901	16.4	295
OTHER	251,638	40.9	736
	615,991		1802 sec (1 hr)

Figure 1

For each job class, the top-ranked (CPU, IO, core) groups were selected until cumulative CPU time exceeded the allotted time for that job class. A synthetic job was then constructed to represent each of the selected groups. The collection of synthetic jobs from all job classes consti-tuted the benchmark mix. This mix was actually run at two service bureaus. However, because of the cost incurred for these first two runs, and because it was felt that this mix did not adequately represent the real workload (since many resource groups were not represented), this approach was abandoned.

It then became clear that with a few reasonable assumptions concerning the nature of the service bureau algorithms, a more accurate, low-cost approach could be developed. This approach proved to be suc-cessful for a number of service bureau algorithms and is now described in more detail.

IV. SELECTED APPROACH

A. Assumptions

The main assumptions specific to the approach taken in this study are the following:

1. CPU time, IO channel activity, and core memory space are the major cost factors for a job--all other miscellaneous charges (MISC) have a secondary effect on total job charges.

8

2. Service bureau charging algorithms are of the general form:

$$\$ = W_1 \cdot CPU + W_2 \cdot IO + W_3 \cdot CPU \cdot Core + W_4 \cdot IO \cdot Core + W_5 \cdot MISC.$$

3. The assigning of jobs to discrete core breakpoints has only a minor effect on total job charges.
4. A synthetic job can be constructed to place a prescribed load on the CPU, IO channels, and core.

Each of these assumptions will be discussed in more detail in the following paragraphs.

B. Theory

The following describes in theoretical terms the selected synthetic benchmark process and its adequacy as a tool for comparing the cost of processing a given workload by different service bureaus. Let $R = \{R_1, \ldots, R_m\}$ be the set of resources to be modeled. Because core, the CPU, and IO channels are assumed to be primary contributors to a job's cost, they were chosen as the set R.

A synthetic job, S, is a program capable of driving each of the m resources in a prescribed manner. The demand which S puts on each of the resources is a function of specific values of the input parameter list $P = (P_1, \ldots, P_q)$. In order that this functional relationship may be empirically determined, S is first calibrated on the base machine over a wide range of values for each P_i. A linear regression model or source code analysis is then used to determine some function, f, between P and the utilization of each resource in set R. For example, assume that a relationship is found between the CPU time and EXCP count for S and the parameter list P:

$$(CPU, EXCP) = f(P).$$

Then, given a prescribed CPU time, t, and a desired EXCP count, d, the program S with parameter list $P = f^{-1}(t,d)$ should execute in CPU time t and should produce d EXCP's.

9

Assume that all jobs in the real workload are partitioned into r core groups:

$$C_1, \; C_2, \ldots, C_r$$

where

$$C_i = C_{i-1} + \Delta$$

is the i'th core breakpoint. For this study, Δ = 40K bytes. A job is assigned to core group C_i if its core size falls within $\pm \frac{\Delta}{2}$ of C_i. Let CPU_{ij} and IO_{ij} be the CPU time and IO channel activity (expressed in EXCP counts) for the j'th job of core group C_i. For each C_i, let:

$$CPU_i = \frac{\sum\limits_{j} CPU_{ij}}{n_i}$$

$$IO_i = \frac{\sum\limits_{j} IO_{ij}}{n_i}$$

represent the CPU time and EXCP counts, respectively, for the "average job," where n_i is the number of jobs in C_i. Let sf_i be the appropriate scaling factor which reduces CPU_i to an arbitrary value, t' (here, t' was chosen to be 10 seconds):

$$sf_i = \begin{cases} 1 \text{ if } CPU_i \leq t' \\ \\ \dfrac{CPU_i}{t'} \text{ if } CPU_i > t' \end{cases}$$

10

Calculate the new scaled averages:

$$CPU_i' = \frac{1}{sf_i} \cdot CPU_i$$

$$IO_i' = \frac{1}{sf_i} \cdot IO_i$$

Finally, determine the appropriate input parameter list $p_i = f^{-1}(CPU_i', IO_i')$ which causes S to exactly duplicate the scaled averages CPU_i' and IO_i'. If S_i represents S forced to a core size of C_i (e.g., via the REGION parameter on the EXEC card), then $S_i(p_i)$ will be used to represent all jobs in core group C_i.

Inspection of the actual service bureau charging algorithms supports the claim that many are of the general form:

$$\$ = W_1 \cdot CPU + W_2 \cdot IO + W_3 \cdot CPU \cdot Core + W_4 \cdot IO \cdot Core + W_5 \cdot MISC$$

where MISC are miscellaneous charges that are assumed to have a secondary effect on the total cost of a job (e.g., disk and tape allocation charges, charges for temporary work space). Note, however, that the following implicit assumptions still remain: 1) no minimum job charges exist in any of the service bureau pricing schemes; and 2) disk and tape channel activity have the same weighting factor (i.e., W_4). The cost of actually running a job J_{ij} with core size $C_i + \delta_{ij}$ ($|\delta_{ij}| \leq \frac{\Delta}{2}$), CPU time CPU_{ij}, EXCP count IO_{ij}, and miscellaneous charges $MISC_{ij}$, is thus:

$$\$_{ij} = W_1 \cdot CPU_{ij} + W_2 \cdot IO_{ij} + W_3 \cdot CPU_{ij} \cdot (C_i + \delta_{ij}) + W_4 \cdot IO_{ij} \cdot (C_i + \delta_{ij})$$
$$+ W_5 \cdot MISC_{ij}$$

11

The actual cost of running all jobs represented by core group C_i is then:

(1)
$$\$actual_i = \sum_j \$_{ij} = W_1 \cdot \sum_j CPU_{ij} + W_2 \cdot \sum_j IO_{ij}$$

$$+ W_3 \cdot C_i \cdot \sum_j CPU_{ij}$$

$$+ W_3 \cdot \sum_j CPU_{ij} \cdot \delta_{ij} + W_4 \cdot C_i \cdot \sum_j IO_{ij}$$

$$+ W_4 \cdot \sum_j IO_{ij} \cdot \delta_{ij} + W_5 \cdot \sum_j MISC_{ij}$$

and the actual cost of all jobs in all core groups is:

$$\$actual = \sum_i \$actual_i$$

Now, assume that the synthetic job $S_i(p_i)$ is run at the service bureau. Its cost will be:

$$\$(S_i) = W_1 \cdot CPU_i' + W_2 \cdot IO_i' + W_3 \cdot CPU_i' \cdot C_i + W_4 \cdot IO_i' \cdot C_i + W_5 \cdot MISC_i'$$

where $MISC_i'$ is primarily due to the synthetic's temporary work space. Expanding the above, we obtain:

12

$$\$(S_i) = W_1 \cdot \frac{1}{sf_i} \cdot CPU_i + W_2 \cdot \frac{1}{sf_i} \cdot IO_i$$

$$+ W_3 \cdot \frac{1}{sf_i} \cdot CPU_i \cdot C_i + W_4 \cdot \frac{1}{sf_i} \cdot IO_i \cdot C_i + W_5 \cdot MISC_i'$$

$$= W_1 \cdot \frac{\sum_j CPU_{ij}}{sf_i \cdot n_i} + W_2 \cdot \frac{\sum_j IO_{ij}}{sf_i \cdot n_i}$$

$$+ W_3 \cdot C_i \cdot \frac{\sum_j CPU_{ij}}{sf_i \cdot n_i} + W_4 \cdot C_i \cdot \frac{\sum_j IO_{ij}}{sf_i \cdot n_i} + W_5 \cdot MISC_i'$$

Let $ef_i = sf_i \cdot n_i$ be the expansion factor for S_i. The estimated cost for all the jobs in core group C_i is then:

$$(2) \qquad \$est_i = ef_i \cdot \$(S_i) = W_1 \cdot \sum_j CPU_{ij}$$

$$+ W_2 \cdot \sum_j IO_{ij}$$

$$+ W_3 \cdot C_i \cdot \sum_j CPU_{ij}$$

$$+ W_4 \cdot C_i \cdot \sum_j IO_{ij}$$

$$+ W_5 \cdot sf_i \cdot n_i \cdot MISC_i'$$

Comparing (1) and (2), this estimate is in error by:

$$\epsilon_i = \$actual_i - \$est_i = W_3 \cdot \sum_j CPU_{ij} \cdot \delta_{ij} + W_4 \cdot \sum_j IO_{ij} \cdot \delta_{ij}$$

$$+ W_5 \cdot (\sum_j MISC_{ij} - sf_i \cdot n_i \cdot MISC_i')$$

The estimated cost of the complete workload W is given by:

$$\$est = \sum_i \$est_i$$

and is in error by:

$$\epsilon = \$actual - \$est = \sum_i \epsilon_i$$

$$= W_3 \cdot \sum_{ij} CPU_{ij} \cdot \delta_{ij} + W_4 \cdot \sum_{ij} IO_{ij} \cdot \delta_{ij}$$

$$+ W_5 \cdot (\sum_{ij} MISC_{ij} - MISC')$$

where $MISC' = \sum_i sf_i \cdot n_i \cdot MISC_i'$. Inspection of the service bureau algorithms and the SMF data indicated that the first two error terms, which are due to the assigning of jobs to discrete core breakpoints, would have a minimal impact on total job charges.

Recall the four major assumptions of the synthetic approach:

1. MISC charges have a secondary effect on total job charges.
2. Service bureau charging algorithms have the general form:

$$\$ = W_1 \cdot CPU + W_2 \cdot IO + W_3 \cdot CPU \cdot Core + W_4 \cdot IO \cdot Core + W_5 \cdot MISC.$$

3. The assigning of jobs to discrete core breakpoints has only a minor effect on total job charges.

14

4. A synthetic job can be constructed to place a prescribed load on the CPU, IO channels, and core.

Assumption 2 appears valid, as has been already noted, for many service bureau algorithms. The impact of minimum charges and the assumption of equal weighting for tape and disk activity in the algorithms are addressed in Section V, Results, as is the validity of Assumption 1. As has been noted, Assumption 3 appears valid from inspection of the service bureau algorithms and the SMF data base. The validity of Assumption 4 is discussed on p. 19.

It is interesting to note that the approach which was finally arrived at closely resembled that suggested by Crowding [5] (in the sense that an "average job" within each core group is computed), although both approaches were formulated independently of each other. In addition, both had completely different purposes, as noted earlier.

C. Procedures

The following is a list of the steps required to generate a synthetic mix for the purposes of this study:

1. Choose an appropriate synthetic program.
2. Calibrate the synthetic on the base machine over a wide range of input parameter values.
3. Determine a functional relationship between the synthetic input parameters and CPU time and EXCP counts.
4. Partition the workload into core groups, each identified by the mid-point of a 40K increment.
5. For each core group, determine the "average" job--i.e., the average CPU time and EXCP count.
6. Scale these averages, if need be, to reduce the CPU time to 10 seconds of base machine time.
7. Generate a synthetic job which runs in the prescribed core size and which duplicates the scaled CPU time and EXCP averages.
8. Combine the jobs from each core group to form the synthetic mix.

Four considerations should be taken into account when choosing the synthetic program. First, the program should be capable of exercising

15

the CPU and IO in a prescribed manner via appropriate input parameters. Second, the program should be portable across a family of operating systems. Third, the program should be easy to instrument in order that its CPU time and IO activity (EXCP counts) may be determined during the calibration phase. Finally, the program's source code should not be subject to varying degrees of optimization at each different site. A slightly modified version of the Fortran synthetic program described by Shetler and Bell [12] and originally developed by Buchholz [4] meets all of these criteria and was therefore chosen as the synthetic program for this study. Appendix A contains a source listing of the program. Note that this program currently models all IO channel activity via disk IO.

In order to calibrate the synthetic with respect to CPU time, it was first instrumented with STIMER and TTIMER calls [13]. A call via the STIMER macro initiated an interval timer which was thenceforth enabled only during the active execution of the program. Succeeding calls via the TTIMER macro allowed for accurate timings of various portions of the synthetic job. A wide range of NMAS (number of master records) and NCPURP (number of CPU loop repetitions per detail record) values were input and the corresponding CPU times on DMC's 370/165 were noted, as shown in Table 1. The NDET (number of detail records) and NPASS (number of passes) parameters were held constant at 12 and 1, respectively. Using the model:

$$\text{CPU time} = X_1 \cdot \text{NMAS} + X_2 \cdot \text{NCPURP} + X_3$$

linear regression analysis of the data in Table 1 produced the following relationship between CPU time and the NMAS and NCPURP parameters:

(3)   CPU time = .005276·NMAS + .000832·NCPURP + .9763375

Simple source-code analysis of the program produced the following relationship between EXCP counts and NMAS values:

(4)   EXCP's = 7·NMAS + 58

16

This count includes 12 EXCP's for the synthetic's SYSIN and SYSOUT activity. Furthermore, the following DCB parameter settings for the four temporary DASD data sets used by the program are assumed:

Unit	BLKSIZE	LRECL	RECFM
7	80	76	VS
8	800	796	VS
9	800	796	VS
10	800	796	VS

From the inverse of (3) and (4), the proper values of NMAS and NCPURP needed for the synthetic to duplicate a prescribed CPU time and EXCP count can be determined:

(5) $NMAS = (EXCP - 58)/7$

(6) $NCPURP = (CPU - .9763375 - .005276 \cdot NMAS)/.000832$

The type 4 (step termination) SMF records were partitioned into the four classes: H, D, J, OTHER. The CPU time, total tape and disk EXCP count, and core-requested fields for each type 4 record were stripped off and written to one of four files, corresponding to the appropriate job class in which the step was executed. Because at the time of data collection the DMC configuration consisted of both a 370/155 and a 370/165, it was necessary to convert all CPU time to a single base. Therefore, all 370/155 CPU time was expressed in equivalent 370/165 time. The data within each of the four classes was then partitioned, for convenience, into the following nine core groups:

Core Group	Midpoint	Range
1	20K	0– 40K
2	60K	40– 80K
3	100K	80–120K
4	140K	120–160K
5	180K	160–200K
6	220K	200–240K
7	260K	240–280K
8	300K	280–320K
9	340K	320–360K

An average CPU time and EXCP count was then found for each core group within each class. Several of the core groups required the averages to be scaled in order to reduce the average CPU time to 10 seconds, a reasonably manageable number. Table 2 lists the number of jobs in each core group and the total and average CPU times and EXCP counts. Table 3 lists the scaled times and counts with appropriate expansion factors. These factors were applied to the cost of each synthetic in order to arrive at an estimated cost of running the total workload. These projected estimates are discussed in Section V, Results.

From the scaled CPU times and EXCP counts in Table 3, the appropriate NMAS and NCPURP synthetic input parameters were calculated using equations (5) and (6). Table 4 lists the synthetic input parameters and the REGION size required to represent each core group. NMAS must be forced to at least a size of 12 for the synthetic to execute properly. It is to be noted that the core group with midpoint 20K was actually represented by a synthetic with REGION = 44K. This was required since the synthetic itself was of size 44K. In any case, most of the service bureaus had a minimum core size of from 60K to 85K words.

The final collection of 34 jobs presented in Table 4 constitutes the synthetic mix which was run at DMC and at each of the six service bureaus. The following steps were required to estimate the total workload cost at each service bureau:

1. Run the mix at the priority which results in normalized job

18

costs i.e., the priority with a corresponding service factor of 1.

2. Apply the appropriate expansion factor to the cost of each job. This represents an estimate of the core group cost.

3. For each class, sum the core group estimates. This sum then represents the total cost of each job class.

A discussion of the results obtained from running the generated mix under these conditions now follows.

The synthetic job mix was run at each of the six service bureaus used in the algorithm approach as well as at the DMC itself. Because only individual job costs were of interest and not total mix thruput or turnaround times, the particular sequence of jobs in the mix was not important. Table 5 lists the expected CPU times and EXCP counts for class H versus the actual times and counts achieved on the DMC 370/165. This data supports the assumption that the synthetic can closely duplicate specified CPU times and EXCP counts. Tables 6-9 list the individual job costs for each core group at DMC and each service bureau. In order to maintain the anonymity of the service bureaus and DMC, no site will be identified by name in Tables 6-10.

V. RESULTS

In order to obtain the total projected cost for each class, the expansion factors listed in Table 3 were applied to the individual synthetic job costs (Tables 6-9). Table 10 contains these projected class totals. Note, these are undiscounted estimates and do not include unit record charges.

Due to the earlier assumption concerning the general form of the charging algorithm, it was expected that there would be a discrepancy between the actual workload costs as determined by the algorithm approach and the projected costs using the synthetic approach. What was not known a priori was the magnitude of these MISC charges. As shown in Table 10 the relative error introduced into the synthetic approach by

ignoring MISC charges ranged from a low of 6.3% for site #4 to a high
of 32.1% for site #6. Because a detailed description of the site
algorithms might compromise their confidentiality, only general comments
will be made concerning the effect of ignoring these MISC charges.

For sites #1, #4, #5, and #7 a large portion of the MISC charges is
due to device mounts and minimum charges which the "actual workload costs"
include but for which the synthetic approach did not account. For these
sites the inaccuracy due to ignoring MISC charges ranged from 6.3% to
13.1%. For sites #2, #3, and #6 which show the higher percent difference
rates, most of the MISC charges are due to charges for temporary work
space and different charging factors for tape and disk EXCP's (recall
that the synthetic approach did not distinguish between tape and disk
EXCP's).

VI.   CONCLUSIONS

The synthetic benchmark approach described here was used to project
direct batch processing charges at a number of service bureaus. The
approach relied on a number of assumptions all of which proved to be
valid, or at least resulted in explainable differences between actual
and projected charges.

Because it was not possible to associate DMC job-step SMF records
with their corresponding HASP initiation records, individual job turn-
around times and in-queue times could not be obtained. Therefore no
attempt was made in either the algorithm approach or in the synthetic
approach to account for the impact of service bureau priority schemes,
which usually result in a discount for deferred processing and a premium
for express processing.

Refinement of the synthetic approach to include temporary work space
and a distinction between tape and disk activity should result in a more
accurate estimate of projected workload costs for service bureaus whose
charges rely heavily on these measures. Even in its present form,
however, the synthetic approach described here provides a reasonably

accurate estimate (within 13.1%) of workload costs for a given class of charging algorithms within the same vendor mainframe family.

VII. ACKNOWLEDGMENTS

The author would like to thank Mr. Richard F. Dunlavey for his insight into some of the problems encountered during the course of this study, and Dr. Robert S. Butler for suggesting the scaled-average approach.

REFERENCES

1. The Office of Management and Budget, "Policies for Acquiring Commercial or Industrial Products and Services for Government Use," Washington, D.C., Circular No. A-76, August 30, 1967, 12p.

2. International Business Machines Corp., "IBM System/360 Operating System: System Management Facilities," Form GC28-6712.

3. Dunlavey, R. F., "Service Workload Cost Estimation: The Algorithm Method," to be published.

4. Buchholz, W., "A Synthetic Job for Measuring System Performance," IBM Systems Journal, 8:4 (1969) 309-318, 6refs.

5. Crowding, Edward F., "A Controllable Synthetic Job Stream for Benchmarking," IBM Systems Development Div., Poughkeepsie, N.Y., Rept. No. TR 00.2173-1, 6 June 72, 13p, 2refs.

6. Kernighan, B. W. and P. A. Hamilton, "Synthetically Generated Performance Test Loads," Bell Labs., Murray Hill, N.J., in Association for Computing Machinery, Proceedings of the SIGME Symposium, February 1973, p. 121-126, 6refs.

7. Sreenivasan, K. and A. J. Kleinman, "On the Construction of a Representative Synthetic Workload," Comm. ACM, 17:3 (March 1974) 127-133, 13refs. Errata, Comm. ACM, 17:10 (October 1974) 591.

8. Sreenivasan, K. and J. E. Esposito, "Experiments with the Burroughs B3500 Computer System Using a Synthetic Workload," The MITRE Corp., Bedford, Mass., Rept. No. MTR-2859, January 1975, 46p, 4refs.

9. King, Carl, "User Develops Procedure to Generate Synthetic Benchmark Jobstream," EDP Performance Review, 2:4 (April 1974) 1-5.

10. Ferrari, Domenico, "Workload Characterization and Selection in Computer Performance Measurement," Computer, 5:4 (July/August 1972) 18-24, 22refs.

11. Morgan, D. E. and J. A. Campbell, "An Answer to a User's Plea?" in Association for Computing Machinery, Proceedings of the SIGME Symposium, February 1973, p. 112-120, 22refs.

12. Shetler, A. C. and T. E. Bell, "Computer Performance Analysis: Controlled Testing," The RAND Corp., Santa Monica, Calif., Rept. No. R-1436-DCA, April 1974, 83p, 16refs.

13. International Business Machines Corp., "IBM System/360 Operating System Supervisor and Data Management Macro Instructions," White Plains, N.Y., Form C28-6647-3, Fifth Edition, June 1970, 330p.

APPENDICES

APPENDIX A
SYNTHETIC PROGRAM LISTING

```
C
C          FROM IBM SYSTEMS JOURNAL, VOL. 8, NO. 4 BY BUCHOLZ
C          TRANSLATEO TO FORTRAN BY J. F. MARANZANO, BELL TEL. LABS
C          MODIFIEO BY T. E. BELL, THE RAND CORP.
C
C
C          THIS PROGRAM IS FOR USE IN GENERATING A SYNTHETIC BATCH STREAM.
C          PARAMETERS ON INPUT CARDS CAUSE THE JOB TO USE THE COMPUTER
C          SYSTEM LIKE A CPU-BOUNO JOB,  N I/O BOUNO JOB, OR A JOB WITH
C          ANY OEGREE OF CPU ANO I/O MIX.  EACH PARAMETER CARD CAUSES
C          ONE OR MORE TIMED PASSES THROUGH THE SPECIFIEO PASS.  AT THE
C          END OF WORK FOR ONE PARAMETER CARD, THE RESULTS ARE PRINTEO
C          SEVERAL PARAMETER CARDS.  ENO OF INPUT IS INOICATED BY A
C          OUT.  OIFFERENT TYPES OF PASSES CAN BE SPECIFIED BY INPUTTING
C          OUMMY PARAMETER CARD WITH NPASS=0.
C
C          THE JOB FIRST INITIALIZES FILES FOR A SET OF PASSES (SPECIFIED
C          BY ONE PARAMETER CARO).  EACH PASS HAS ITS OWN INITIALIZATION
C          ANO TERMINATION TO REWINO FILES ANO TO SUMMARIZE RESULTS.
C          HOWEVER, OUTPUT IS HELO UNTIL ALL PASSES OF THE PARAMETER CARO
C          ARE COMPLETED.  THIS ELIMINATES SYSOUT BEING OVERLAPPEO WITH
C          TESTS.
C
C          I/O CAN BE OIRECTEO TO TAPE, DISC, OR DRUM.  REAOING ANO
C          WRITING ARE UNFORMATTEO TO REDUCE FORTRAN CONVERSION ROUTINE
C          EXECUTION.  I/O UNITS ARE USEO AS FOLLOWS -
C          1.   MASTER FILE WRITTEN OURING PASS SET INITIALIZATION.  IT IS
C               SUBSEQUENTLY REAO FOR WRITING TO FILE 4.
C          2.   OETAIL FILE WRITTEN OURING PASS SET INITIALIZATION.  IT IS
C               SUBSEQUENTLY READ FOR WRITING TO FILE 3.
C          3.   OETAIL FILE WRITTEN OURING PASS FROM FILE 2.
C          4.   MASTER FILE WRITTEN OURING PASS FROM FILE 1.
C
C          IF A PASS HAS MASTER RECOROS, ONE OR MORE MASTER RECORDS ARE
C          PROCESSEO FOR EACH OETAIL RECORD.  EACH DETAIL RECORD PROCESSED
C          IS FOLLOWEO BY THE NUMBER OF CPU REPETITIONS INOICATED ON THE
C          PARAMETER CARO.  IF NOET = 0, NO CPU REPETITIONS WILL BE
C          PERFORMEO.  HOWEVER, IF NCPURP = 0, OETAIL RECOROS WILL BE
C          PROCESSEO IF REQUESTED.
C
C          IF A PASS HAS NO MASTER RECOROS, NCUPRP REPETITIONS OF THE
C          CPU LOOP WILL BE PERFORMEO.
C
C
C          THE PROGRAM IS WRITTEN IN FORTRAN, ANO CALLS A ROUTINE NAMEO
C          ZTIME TO OBTAIN THE CURRENT COMPUTER CLOCK VALUE.  THIS
C          ROUTINE MUST BE WRITTEN FOR EACH SYSTEM.
C
C
C          INPUT
C               NMAS      NUMBER OF MASTER RECOROS.  NMAS MUST BE GREATER
C                         THAN ZERO FOR ANY I/O TO BE OONE.
C               NOET      NUMBER OF DETAIL RECORDS (LESS THAN OR EQUAL TO
C                         NMAS)
C               NCPURP    NUMBER OF REPETITIONS OF CPU MINOR LOOP FOR
```

```
C                       EACH DETAIL RECORD.  IF NO DETAIL OR MASTER
C                       RECORDS, NCPUR IS THE TOTAL NUMBER OF
C                       REPETITIONS PER PASS IN TOTALLY CPU-BOUND
C                       PASSES.
C             NPASS     NUMBER OF IDENTICAL PASSES THROUGH LOOP DEFINED
C                       BY NMASS, NDET, AND NCPURP
C
C           THE FORMAT OF THE PARAMETER INPUT CARD IS AS SHOWN BELOW.
C           THE ZEROS WOULD, OF COURSE, BE REPLACED WITH THE DESIRED
C           VALUES.  THE VARIABLE IDENTIFIERS ARE PRESENT ONLY FOR
C           CONVENIENCE.  (PARAMETER CARDS ARE READ UNDER FORMAT 88.)
C
C234567890123456789012345678901234567890123456789n12345678901234567890
CNMAS=0000000,NDET=0000000,NCPURP=0000000,NPASS=00
C
C
C
C           THE MASTER FILE AND DETAIL FILE ARE DEFINED HERE
C           BY THE LENGTH OF THE 'RECORD' VARIABLE WHICH WILL BE WRITTEN.
C           'RECORD' MUST BE DIMENSIONED AS RECORD(N) WITH N 1 OR MORE.
0001        INTEGER RECORD (47)
0002        INTEGER START, SUM, TABLE(1000)
0003        INTEGER X1,X2,X3,X4,X5,X6
0004        DIMENSION TIMES(100,2)
0005        DATA N/10/,START/100/
C
0006        CALL BTIME
0007        WRITE(6,87)
C
C           READ INPUT PARAMETERS
C
0008   1000 CALL ZTIME (TIMEN)                                          ********
0009        READ(5,88) NMAS, NDET, NCPURP,NPASS                              *
C           FORCE NMAS TO AT LEAST 12
0010        IF (NMAS.LT.12) NMAS=12
0011        WRITE(6,89) NMAS, NDET, NCPURP, NPASS                            *
C                                                                           *
C           CHECK FOR UNUSUAL INPUT                                         *
C                                                                           *
C           END OF INPUT CARDS                                              *
0012        IF (NPASS .LE. 0) CALL EXIT                                      *
C           MORE DETAIL THAN MASTER  --  DISREGARD                          *
0013        IF(NMAS .GE. NDET) GO TO 30                                      *
0014        WRITE(6,90)                                                      *
0015        GO TO 1000                                                      *
C           REQUESTED CPU REPETITIONS, BUT NDET = 0, NMAS GT 0              *
0016     30 IF((NMAS .LE. 0) .OR. (NDET .GT. 0) .OR. (NCPURP .LE. 0)) GO TO 32   *
0017        WRITE(6,91)                                                      *
0018     32 CONTINUE                                                        *
C                                                                           *
C           END OF PARAMETER INPUT PROCESSING                               *
C                                                                           *
C           INITIALIZE TABLE WITH NUMBERS                                   *
0019        K = N**3                                                        *
0020        DO 200 I=1,K                                             P       *
```

26

```
0021        200     TABLE(I) = START+I-1                                        A  S  *
0022                MKEY = 0                                                    S  E  *
0023                LKEY = 0                                                    S  T  *
        C                                                                          U  *
        C                                                                       S  P  *
0024                IF(NMAS .EQ.0) GO TO 280                                    E     *
        C           MASTER GENERATION                                           T     *
0025            RECORD(1) = 0                                                         *
0026                DO 250 MKEY=1,NMAS                                                *
0027                RSUM = 0.                                                         *
0028                MCHK = MKEY                                                       *
0029        250     wRITE (7) MKEY, RSUM, MCHK, RECORD                                *
0030                RECORD(1) = 1                                                     *
0031                wRITE (7) MKEY, RSUM, MCHK, RECORD                                *
0032                RECORD(1) = 0                                                     *
0033                ENDFILE 7                                                         *
        C                                                                             *
        C                                                                             *
0034        251     IF(NOET .EQ. 0)GO TO 280                                          *
        C           DETAIL GENERATION                                                 *
0035                NRATO = NMAS/NOET                                                 *
0036                DO 275 LKEY = NRATO,NMAS,NRATO                                    *
0037                DSUM =0.                                                          *
0038                LCHK = LKEY                                                       *
0039        275     wRITE (8) LKEY, DSUM, LCHK, RECORD                                *
0040                RECORD(1) = 1                                                     *
0041                wRITE (8) LKEY, DSUM, LCHK, RECORD                                *
0042                RECORD(1) = 0                                                     *
0043                ENDFILE 8                                                         *
        C                                                                    ********
        C
        C
0044        280   CONTINUE
        C
        C
        C           DO LOOP NPASS TIMES                                        ********
        C                                                                             *
0045                DO 1 ICNT = 1, NPASS                                       P  S  *
        C                                                                       A  E  *
        C           IF NO MASTER OR DETAILED RECORDS, SKIP I/O SETUP           S  T  *
0046                IF(NMAS .EQ. 0) GO TO 285                                   S  U  *
0047                KCNT = 0                                                    P     *
0048                REWIND 10                                                         *
0049                REWIND 7                                                          *
0050                READ (7) MKEY, RSUM, MCHK, RECORD                                 *
0051                IF(NOET .GT. 0) GO TO 283                                         *
0052                LKEY = 999999                                                     *
0053                GO TO 285                                                         *
0054        283     REWIND 9                                                          *
0055                REWIND 8                                                          *
0056                READ (8) LKEY, DSUM, LCHK, RECORD                         ********
        C
        C
        C           START TIMED PASS            ***********************************************
        C
```

27

```
0057        285     CALL ZTIME(FSTRTM)
0058                TIMES(ICNT, 1) = FSTRTM
            C
            C
0059        400        IF(MKEY-LKEY)   475,450,425
            C           SEQUENCE ERROR
0060        425     WRITE(6,94)
0061                STOP    6
            C           IF NOT REQUESTED, SKIP THE COMPUTE KERNEL
0062        450     IF(NCPURP .LE. 0) GO TO 301
            C
            C
            C           COMPUTE KERNEL
            C
            C           REPEAT NCPUR TIMES FOR EACH DETAIL RECORD (OR JUST NCPUR IF
            C           NMAS = NDET = 0)
            C
            C           START CPU LOOP                                          *********************
0063                DO 300 1 = 1, NCPURP
0064                SUM = 0
0065                IU = 0
0066                J = 0
0067                DO 350 K=1,N
0068                J = J+(6*IU+1)
0069                SUM = SUM + TABLE(J)
0070        350     IU =IU +K
            C                                                                   CPU LOOP
0071                LSUM = (N*(N+1))/2
0072                IF(START .EQ. (SUM-LSUM*LSUM)/N+1)GO TO 300
0073                WRITE(6,96)
0074                STOP    5
0075        300     CONTINUE
            C           END OF CPU LOOP                                         *********************
            C
            C           IF NO I/O REQUESTED, SKIP I/O SECTION
0076        301     IF(NMAS .EQ. 0) GO TO 600
            C
            C
            C           START OF I/O SECTION                                    *********************
            C
0077        460     CONTINUE
            C           WRITE DETAIL RECORDS
0078                RSUM = SUM
0079                DSUM = SUM
0080                ICHK = LCHK
0081                KCNT = KCNT+1
0082                WRITE (9) LKEY, DSUM, LCHK, RECORD
0083                READ (8) LKEY, DSUM, LCHK, RECORD
0084                IF(RECORD(1) .EQ. 1) GO TO 500
            C
            C           WRITE MASTER
            C                                                                   I/O LOOP
0085        475     WRITE (10) MKEY, RSUM, MCHK, RECORD
0086                READ (7) MKEY, RSUM, MCHK, RECORD
0087                IF(RECORD(1) .EQ. 1) GO TO 600
```

28

```
0088              GO TO 400
        C
        C          NO DETAILED RECORDS LEFT
        C
0089    500       LKEY = 999999
0090              GO TO 475
        C
        C          NO MASTER RECORDS LEFT
        C
0091    600       CONTINUE
        C          GET END TIME
0092              CALL ZTIME(FENDTM)
0093              TIMES (ICNT, 2) = FENDTM
        C
        C          END OF I/O SECTION                           ********************
        C
        C
        C          END OF TIMED PASS           *************************************
        C                                                                   ********
        C          IF I/O SECTION HAS BEEN USED, CHECK FOR CORRECT OPERATION    P C *
0094              IF((NMAS.EQ.0) .OR. (NDET.EQ.0)) GO TO 1                      A H *
0095              IF(ICHK - (KCNT*NRATO))700,1,700                             S E *
0096    700       WRITE(6,93) ICHK,ICNT,NRATO                                   S C *
        C                                                                       K *
        C                                                                   ********
0097    1         CONTINUE
        C
        C
        C                                                                   ********
        C                                                                       R *
        C          PUT OUT TOTAL ELAPSED TIME FOR THIS SET OF PASSES           F E *
        C                                                                       I S *
        C                                                                       N U *
0098              CALL ZTIME(TIMEF)                                            A L *
0099              DIFF = TIMEF - TIMEN                                         L T *
0100              WRITE(6,95) TIMEN, TIMEF, DIFF                               S *
        C                                                                   ********
        C
        C          COMPUTE RESULTS                                          ********
        C                                                                       *
0101              WRITE(6,97)                                                   *
0102              STIMES = 0.                                                   *
0103              SSTIME = 0.                                                   *
0104              DO 1100 ICNT = 1, NPASS                                       *
0105              TIMEON = TIMES(ICNT, 1)                                       *
0106              TIMEOF = TIMES(ICNT, 2)                                       *
0107              DIFF = TIMEOF - TIMEON                                        *
0108              WRITE(6,99) TIMEON, TIMEOF, DIFF                             R *
0109              STIMES = STIMES + DIFF                                        E *
0110              SSTIME = SSTIME + DIFF ** 2                                  P S *
0111    1100      CONTINUE                                                     A U *
0112              TMEAN = STIMES / FLOAT(NPASS)                               S L *
0113              TSTDEV = SQRT((SSTIME / FLOAT(NPASS)) - TMEAN ** 2)         S T *
0114              WRITE(6,98) TMEAN, TSTDEV                                    S *
        C                                                                       *
```

29

```
          C          GO GET NEXT CONTROL CARD                                          *
          C                                                                            *
0115               REWIND 7                                                            *
0116               REWIND 8                                                            *
0117               REWIND 9                                                            *
0118               REWIND 10                                                           *
0119               GO TO 1000                                                          *
          C                                                                      *********
          C     FORMAT STATEMENTS
          C
0120      87   FORMAT(24H1 SYNTHETIC BATCH STREAM)
0121      88   FORMAT(6X, I7, 6X, I7, 8X, I7, 7X, I2)
0122      89   FORMAT(10H0 MASTER =,I8,4X,8HDETAIL =,I8,4X,17HCPU REPETITIONS =,
               1    I8,4X,8HPASSES =,I5)
0123      90   FORMAT(38H  MORE DETAILS THAN MASTERS SPECIFIED.)
0124      91   FORMAT(30H  NO CPU LOOPS SINCE NDET = 0.)
0125      93   FORMAT(23H CHECKSUM ERROR, ICHK= ,I6,7H KCNT= ,I6,8H IRATO= ,I6)
0126      94   FORMAT(15H SEQUENCE ERROR)
0127      95   FORMAT(10H0ON-TIME =,F10.3,4X,10HOFF-TIME =,F10.3,
               1    4X,20HTOTAL ELAPSED TIME =,F10.3)
0128      96   FORMAT(14H COMPUTE ERROR)
0129      97   FORMAT(1H0,11X,5HSTART,11X,3HEND,12X,12HELAPSED TIME)
0130      98   FORMAT(1H ,22X,19HMEAN ELAPSED TIME =,F10.3,
               1    22H   STANDARD DEVIATION =,F10.3//)
0131      99   FORMAT(1H , 8X, F10.3,  6X, F10.3,  7X, F10.3)
0132           END
```

```
0001            SUBROUTINE ZTIME (TIMET)
        C
        C       RETURNS CPU ELAPSED TIME IN SECONDS
        C       USED ONLY DURING THE CALIBRATION PHASE -- HENCE A DUMMY ITIME
        C       FOR THE RUNNING VERSION OF THE SYNTHETIC
        C
0002            DUMMY=0.0
        C       ITIME RETURNS NO. OF TIMER UNITS REMAINING (1 TIMER UNIT=26 USEC)
0003            I=ITIME(DUMMY)
        C       STIMER SETTING WAS 16**4/100
0004            TIMET=16.0**4/100.0-26.0*(I*.000001)
0005            RETURN
0006            END



0001            SUBROUTINE BTIME
        C       DUMMY BTIME
0002            I=1
0003            RETURN
0004            END



0001            FUNCTION ITIME (DUMMY)
        C       DUMMY ITIME
0002            ITIME=0
0003            RETURN
0004            END
```

31

APPENDIX B
TABLES

TABLE 1

SYNTHETIC PROGRAM CALIBRATION

(NDET = 12, NPASS = 1)

NMAS	NCPURP	DMC 370/165 Time (sec)	NMAS	NCPURP	DMC 370/165 Time (sec)
12	0	.765	12	4000	3.757
50	0	1.005	50	4000	3.827
100	0	1.201	100	4000	4.456
500	0	3.541	500	4000	5.721
900	0	5.837	900	4000	8.337
1000	0	6.503	1000	4000	8.939
5000	0	32.239	5000	4000	32.764
9000	0	50.256	9000	4000	50.199
10000	0	53.161	10000	4000	53.035
12	500	1.015	12	16000	14.530
50	500	1.201	50	16000	15.372
100	500	1.561	100	16000	14.873
500	500	3.401	500	16000	17.801
900	500	5.491	900	16000	19.682
1000	500	5.987	1000	16000	20.204
5000	500	28.644	5000	16000	41.703
9000	500	50.173			
10000	500	53.986			
12	1000	1.528	12	90000	75.672
50	1000	1.787	50	90000	76.574
100	1000	1.937	100	90000	76.231
500	1000	4.247	500	90000	79.639
900	1000	7.319	900	90000	79.153
1000	1000	7.105			
5000	1000	29.586			
9000	1000	51.677			
10000	1000	50.915			

TABLE 2

CORE GROUP AVERAGES

Class	Group	No. of Jobs	Total 165 CPU Time (sec)	Total EXCP Count	Average CPU Time	Average EXCP Count
H	1.	5538	5538	554100	1	100
	2.	5819	35919	22452704	6	3858
	3.	2970	61066	23538768	20	7925
	4.	619	30351	9760500	49	15768
	5.	907	32452	10649700	35	11741
	6.	175	10000	1474500	57	8425
	7.	57	18912	1628200	331	28564
	8.	13	372	112600	28	8661
	9.	26	4698	1514400	180	58246
D	1.	5006	5254	701600	1	140
	2.	10103	16254	4933400	1	488
	3.	12253	33275	6731700	2	549
	4.	3169	17912	1766100	5	557
	5.	1958	9156	1502000	4	767
	6.	154	635	94200	4	611
	7.	11	344	2200	31	200
	8.	31	312	9800	10	316
	9.	4	340	20000	85	5000
J	1.	5536	6203	744000	1	134
	2.	6782	21810	10295300	3	1518
	3.	8014	49324	13781100	6	1719
	4.	1362	12035	2118600	8	1555
	5.	799	17597	2977600	22	3726
	6.	51	866	64400	16	1262
	7.	16	166	7300	10	456
	8.	1	6	100	6	100
OTHER	1.	2209	2627	485500	1	219
	2.	6562	33253	20882448	5	3182
	3.	4071	67709	22316256	16	5481
	4.	1270	38334	6530300	30	5141
	5.	652	65851	6040800	100	9265
	6.	205	29671	1483700	144	7237
	7.	43	19431	503400	451	11706
	8.	6	6	800	1	133

TABLE 3

CORE GROUP SCALED AVERAGES AND
EXPANSION FACTORS

Class	Core Group	Scaled Avg. CPU Time (sec)	Scaled Avg. EXCP Count	Expansion Factor
H	1.	1	100	5538
	2.	6	3858	5819
	3.	10	3963	5940
	4.	10	3218	3033.1
	5.	10	3355	3174.5
	6.	10	1478	997.5
	7.	10	863	1886.7
	8.	10	3093	36.4
	9.	10	3236	468
D	1.	1	140	5006
	2.	1	488	10103
	3.	2	549	12253
	4.	5	557	3169
	5.	4	767	1958
	6.	4	611	154
	7.	10	65	34.1
	8.	10	316	31
	9.	10	588	34
J	1.	1	134	5536
	2.	3	1518	6782
	3.	6	1719	8014
	4.	8	1555	1362
	5.	10	1694	1757.8
	6.	10	789	81.6
	7.	10	456	16
	8.	6	100	1
OTHER	1.	1	219	2209
	2.	5	3182	6562
	3.	10	3426	6513.6
	4.	10	1714	3810
	5.	10	927	6520
	6.	10	503	2952
	7.	10	260	1939.3
	8.	1	133	6

TABLE 4

SYNTHETIC MIX SPECIFICATION

Class	Core Group	Synthetic Input Parameters (NDET=12,NPASS=1) NMAS	NCPURP	REGION Size (K)
H	1.	12	0	44
	2.	543	2595	60
	3.	558	7307	100
	4.	451	7986	140
	5.	471	7859	180
	6.	203	9558	220
	7.	115	10116	260
	8.	434	8094	300.
	9.	454	7967	340
D	1.	12	0	44
	2.	61	0	60
	3.	70	786	100
	4.	71	4386	140
	5.	101	2994	180
	6.	79	3133	220
	7.	12	10839	260
	8.	37	10611	300
	9.	76	10364	340
J	1.	12	0	44
	2.	209	1107	60
	3.	237	4535	100
	4.	214	7085	140
	5.	234	9362	180
	6.	104	10186	220
	7.	57	10484	260
	8.	12	6000	300
OTHER	1.	23	0	44
	2.	446	2008	60
	3.	481	7796	100
	4.	237	9343	140
	5.	124	10059	180
	6.	64	10440	220
	7.	29	10662	260
	8.	12	0	300

TABLE 5

EXPECTED VS. ACTUAL CPU TIMES AND EXCP COUNTS
FOR CLASS H ON THE DMC 370/165

Core Groups	Expected CPU Time (sec)	Expected EXCP's	Actual CPU Time (sec)	Actual EXCP's
1.	1	100	.89	142*
2.	6	3858	4.92	3859
3.	10	3963	8.49	3964
4.	10	3218	9.16	3215
5.	10	3355	9.18	3355
6.	10	1478	9.78	1479
7.	10	863	9.55	863
8.	10	3093	9.59	3096
9.	10	3236	9.39	3236

*due to NMAS forced to 12

TABLE 6

INDIVIDUAL SYNTHETIC JOB COSTS ($)

CLASS H

Site Number

Core Group	#1	#2	#3	#4	#5	#6	#7
1.	.95	.29	.92	1.00	.63	.40	.66
2.	8.57	6.73	6.78	6.22	8.70	3.32	5.06
3.	11.17	8.64	12.09	8.04	10.61	5.81	10.73
4.	10.95	10.15	12.94	7.61	10.87	6.69	14.83
5.	12.37	13.53	15.32	8.14	11.64	7.31	22.31
6.	9.17	8.86	11.48	6.66	11.47	7.86	17.35
7.	8.44	7.49	11.23	6.39	13.64	8.85	16.13
8.	15.33	21.09	20.50	8.74	14.30	9.89	49.19
9.	16.96	24.80	23.13	9.38	16.25	10.59	63.87

TABLE 7

INDIVIDUAL SYNTHETIC JOB COSTS ($)

CLASS D

Site Number

Core Group	#1	#2	#3	#4	#5	#6	#7
1.	.95	.29	.92	1.00	.83	.41	.66
2.	1.58	.85	1.35	1.00	1.19	.50	1.27
3.	2.30	1.24	2.33	1.21	1.94	1.34	2.45
4.	4.05	2.41	5.32	2.74	5.22	3.67	4.68
5.	4.29	3.41	5.01	2.37	5.09	2.94	7.01
6.	4.53	3.61	5.56	2.45	4.62	3.47	8.52
7.	6.61	4.02	8.18	5.21	11.57	8.73	9.17
8.	7.67	5.57	9.28	5.70	12.60	9.43	13.51
9.	9.15	8.06	10.99	6.31	13.71	10.37	20.47

TABLE 8

INDIVIDUAL SYNTHETIC JOB COSTS ($)

CLASS J

Site Number

Core Group	#1	#2	#3	#4	#5	#6	#7
1.	.94	.29	.92	1.00	.72	.43	.66
2.	3.81	2.69	3.24	2.51	3.20	1.51	2.33
3.	5.74	3.93	6.35	3.94	6.10	3.12	5.68
4.	6.91	5.46	8.61	4.78	8.10	5.01	8.67
5.	8.80	7.97	11.35	6.11	10.90	6.87	13.79
6.	7.87	6.24	10.13	5.85	11.77	8.61	12.51
7.	8.11	6.06	10.44	6.29	13.50	9.83	13.07
8.	5.13	3.03	5.99	3.26	7.59	5.59	9.12

40

TABLE 9

INDIVIDUAL SYNTHETIC JOB COSTS ($)

CLASS OTHER

Site Number

Core Group	#1	#2	#3	#4	#5	#6	#7
1.	1.14	.47	1.05	1.00	1.23	.31	.71
2.	7.15	5.54	5.68	5.01	5.40	2.97	4.30
3.	10.21	7.64	10.81	7.28	10.31	5.90	9.93
4.	8.02	6.24	9.71	5.90	10.07	6.18	9.87
5.	7.15	5.39	9.59	5.31	10.45	6.97	9.82
6.	6.91	4.88	8.91	5.33	10.73	8.05	9.87
7.	7.68	5.14	9.59	6.10	14.75	10.07	11.03
8.	2.53	1.02	2.20	1.00	1.01	.78	6.76

TABLE 10

PROJECTED WORKLOAD COSTS ($) – SYNTHETIC APPROACH

Class	Site #1	Site #2	Site #3	Site #4	Site #5	Site #6	Site #7
H	227,527	201,170	248,454	161,819	232,360	129,422	292,060
D	71,607	40,687	75,259	44,205	68,420	42,402	77,451
J	102,701	73,399	110,633	71,966	105,950	57,442	102,263
OTHER	228,422	170,453	254,439	167,174	272,093	170,913	246,663
TOTALS	630,257	485,709	688,785	445,164	678,823	400,179	718,437

ACTUAL WORKLOAD COSTS ($) – ALGORITHM APPROACH

	722,971	694,943	961,172	475,234	781,437	589,241	798,565

ESTIMATE OF MISC CHARGES (ACTUAL – PROJECTED)

	92,714	209,234	272,387	30,070	102,614	189,062	80,128

PERCENT DIFFERENCE (MISC/ACTUAL)

	12.8%	30.1%	28.3%	6.3%	13.1%	32.1%	10.0%

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA SHEET**	1. PUBLICATION OR REPORT NO. *NBS TN-920*	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE Use of Synthetic Benchmarks for Estimating Service Bureau Processing Charges		5. Publication Date July 1976	
		6. Performing Organization Code	
7. AUTHOR(S) Dennis M. Conti		8. Performing Organ. Report No.	
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		10. Project/Task/Work Unit No. 640.1135	
		11. Contract/Grant No.	
12. Sponsoring Organization Name and Complete Address *(Street, City, State, ZIP)* Data Management Center Department of Health, Education, and Welfare Washington, D.C. 20203		13. Type of Report & Period Covered Final	
		14. Sponsoring Agency Code	

15. SUPPLEMENTARY NOTES

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

This report describes the development of a new synthetic benchmark technique for estimating batch processing charges at service bureau sites. This technique was used to estimate the cost of processing a large batch workload at a number of service bureaus within the same mainframe family. The method was found to be low-cost, yet reasonably accurate for a certain class of service bureau charging algorithms. Refinements of this method are suggested which will extend its applicability to other algorithms. The procedures used to create and run the benchmark, together with the projection of total workload cost are described.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Benchmarking; charging algorithms; service bureaus; synthetic benchmarking; workload characterization.

18. AVAILABILITY [X] Unlimited	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PAGES
[ ] For Official Distribution. Do Not Release to NTIS	UNCLASSIFIED	47
[X] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, <u>SD Cat. No. C13.46:920</u>	20. SECURITY CLASS (THIS PAGE)	22. Price $1.15
[ ] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151	UNCLASSIFIED	

USCOMM-DC 29042-P74

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH reports National Bureau of Standards research and development in physics, mathematics, and chemistry. It is published in two sections, available separately:

• Physics and Chemistry (Section A)

Papers of interest primarily to scientists working in these fields. This section covers a broad range of physical and chemical research, with major emphasis on standards of physical measurement, fundamental constants, and properties of matter. Issued six times a year. Annual subscription: Domestic, $17.00; Foreign, $21.25.

• Mathematical Sciences (Section B)

Studies and compilations designed mainly for the mathematician and theoretical physicist. Topics in mathematical statistics, theory of experiment design, numerical analysis, theoretical physics and chemistry, logical design and programming of computers and computer systems. Short numerical tables. Issued quarterly. Annual subscription: Domestic, $9.00; Foreign, $11.25.

DIMENSIONS/NBS (formerly Technical News Bulletin)—This monthly magazine is published to inform scientists, engineers, businessmen, industry, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on the 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, $9.45; Foreign, $11.85.

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 world-wide program coordinated by NBS. Program under authority of National Standard Data Act (Public Law 90-396).

NOTE: At present the principal publication outlet for these 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. N. W., Wash. D. C. 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 purpose of the standards is to establish nationally recognized requirements for products, and to 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.

Federal Information Processing Standards Publications (FIPS PUBS)—Publications in this series collectively constitute the Federal Information Processing Standards Register. 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).

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.

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 Service (Springfield, Va. 22161) in paper copy or microfiche form.

Order NBS publications (except NBSIR's and Bibliographic Subscription Services) from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:
Cryogenic Data Center Current Awareness Service

A literature survey issued biweekly. Annual subscription: Domestic, $20.00; foreign, $25.00.

Liquefied Natural Gas. A literature survey issued quarterly. Annual subscription: $20.00.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $20.00. Send subscription orders and remittances for the preceding bibliographic services to National Bureau of Standards, Cryogenic Data Center (275.02) Boulder, Colorado 80302.

AMERICAN REVOLUTION BICENTENNIAL
1776-1976

75 YEARS
NBS
1901-1976