BEST/1™—Design of a tool for computer system capacity planning

by J. P. Buzen, R. P. Goldberg, A. M. Langer, E. Lentz,* H. S. Schwenk, D. A. Sheetz and A. Shum**

BGS Systems, Inc.
Lincoln, Massachusetts

INTRODUCTION

This paper presents the design philosophy used in the BEST/1 computer performance evaluator. Essentially, BEST/1 is a modeling tool which has been developed to address the "what if" questions that arise when evaluating computer system performance. These questions are of importance in such areas as capacity planning, performance tuning, hardware vendor selection and new system design.

From a design standpoint, the most significant features of BEST/1 are those which relate to ease of use. The following five objectives were considered most essential in this regard:

- The analyst who is familiar with the basic terms and concepts of computer performance evaluation should require no special training in order to use BEST/1. That is, the internal algorithms and procedures used by BEST/1 should be insulated from the user to the maximum extent possible.
- The level of effort required to set up and evaluate a BEST/1 model should be commensurate with the analyst's objectives. First order models should be easy to set up and evaluate, with more refined models requiring progressively more effort in terms of data collection and analysis.
- The input data required by BEST/1 should be easy to understand and obtain. Detailed measurements of internal system activity should not be necessary when studying simple questions.
- BEST/1 should operate interactively so that an analyst can have the benefit of working through a problem and exploring a number of related "what if" questions during a single analysis session. The speed requirements associated with interactive operation are also important in sensitivity studies where a wide range of parameter settings have to be explored.
- BEST/1 should generate a set of output reports which are easy to understand and apply.

In order to achieve these objectives, the BEST/1 design is based on an approach which is commonly employed in applications-oriented higher level languages. That is, a user visible front end has been developed which enables analysts to specify models and carry out analyses using concepts and terms which are generally familiar to individuals working in the area of computer performance evaluation. This user visible front end is then automatically translated into an internal BEST/1 analysis model which is insulated from the user in much the same way as machine language programs are insulated from the user of any higher level language. Thus, the most important design features of BEST/1 are those which relate to its user visible front end. These features are discussed in detail in the remainder of this paper.

It should be noted that, in order to meet the objective of interactive operation, the internal BEST/1 analysis models are based on extensions to the theory of multiple class queueing network models. However, the BEST/1 user need not be concerned with the mathematical details of this theory, just as the user of an applications-oriented higher level language need not be concerned with the details of a particular machine language. For this reason, the mathematical foundations of BEST/1 will not be emphasized in this paper. For discussions of the theory, see the references listed in the bibliography.

BEST/1 MODELING CONCEPTS

BEST/1 is an analysis tool which is capable of calculating the performance of almost any computer system on the basis of parameters which characterize that system's hardware, software, and workload. The performance factors which are calculated by BEST/1 include throughput, response time, queue length, processor utilization, and so on. See Figure 1.

The performance calculation function performed by BEST/1 is similar to the function performed by a benchmark experiment. That is, a BEST/1 analysis and a benchmark experiment both enable the analyst to evaluate the way a computer system will perform under a specific set of conditions. However, in the case of a benchmark experiment
In the case of time sharing workloads, the analyst specifies the number of terminals and the average “think time” (i.e., the average time between the system’s response to a given transaction and the user’s initiation of the next transaction). The principal performance factors calculated by BEST/l for these workloads are the response time per transaction and the throughput rate in transactions per hour.

TRANSACTION PROCESSING WORKLOADS

Transaction processing workloads are similar to interactive workloads in that the external load is assumed to be generated by users at terminals. However, instead of specifying the external load in terms of number of terminals and think time, the analyst simply specifies a total arrival rate in “transactions per hour” as illustrated in Figure 4. Most analysts find it convenient to characterize transaction processing workloads in this manner.

Since the arrival rate is specified as an input parameter, and since throughput is equal to arrival rate (as long as system capacity is not exceeded), the most important performance value computed by BEST/l for transaction processing workloads is the response time. BEST/l also calculates the maximum attainable throughput rate in cases where this value is less than the arrival rate specified in the input file.

BATCH PROCESSING WORKLOADS

In the case of batch processing workloads, analysts are usually most interested in system performance during time periods when the input queues are backlogged (i.e., when

The general nature of interactive or time sharing workloads is illustrated in Figure 3. Essentially, each time sharing workload is associated with a set of terminals. The users at the terminals are assumed to generate transactions which arrive at the system and undergo processing. When a transaction has been completed, a response is printed on the user’s terminal. The user then generates a new transaction and the cycle repeats.
there is always at least one job waiting to be loaded into main memory). BEST/1 enables the analyst to specify batch workloads which have this “backlog” property. Schematically, these workloads are represented as shown in Figure 5. The closed loop in this figure indicates that, in a backlog situation, new jobs are loaded into main memory whenever an active job completes and memory space becomes available. Both throughput and response time (turnaround time) are computed for these workloads.

TOTAL NUMBER OF WORKLOADS

The total number of workloads which may be included in a single BEST/1 model is not limited to three. For example, a single BEST/1 model could include two transaction processing workloads, three batch processing workloads, and two time sharing workloads. Alternatively, a model could consist solely of one time sharing, one batch processing, or one transaction processing workload. In Release 4.0 the total number of workloads which may be included in a single BEST/1 model is equal to 9.

JOBS AND TRANSACTIONS

Each of the basic workload types discussed above makes use of the concept of a “job” or a “transaction”. Essentially, these terms both refer to a unit of work which arrives at a system from an external source, undergoes a certain amount of processing, and is eventually completed. Throughput is expressed in jobs or transactions per hour, and response time (or turnaround time) is expressed on a per job or per transaction basis.

It is clear that the terms job and transaction refer to the same essential concept; however, the former term is more commonly used for batch processing workloads, whereas the latter is more common in the case of time sharing and transaction processing. In the following discussion of BEST/1, the two terms will be regarded as being interchangeable.

However, the term “transaction” will be favored in most cases and should be understood to refer to either a batch job or an on-line transaction.

SPECIAL WORKLOADS

By suitable selection and adjustment of parameters, the three basic workload types defined above can be used to represent a number of special workloads which are of importance in evaluating particular systems. These workloads include periodic real time requests, system overhead activities, and job streams with deadlines and critical paths.

TREATMENT OF MULTIPROGRAMMING

Figures 3-5 utilize a single box (SYSTEM) to represent the location of a transaction from the instant it arrives at a system to the instant it departs. The activity which takes place between these two points is described in greater detail in Figure 6.

The interpretation of Figure 6 is straightforward. Small rectangles indicate the locations of queues, and circles represent servers such as CPU’s or I/O devices. As indicated in the diagram, an arriving transaction may at first have to wait in the memory queue until space becomes available. Once it is loaded into memory, the transaction is placed in the CPU queue where it waits until a processor becomes available. The transaction then receives a burst of CPU processing which terminates when an I/O request is generated. At this point, the transaction proceeds to the appropriate I/O device queue. When it reaches the head of the queue associated with that device, an I/O transfer is initiated. Upon completing this transfer, the transaction returns to the CPU queue. The alternating cycle of CPU bursts and I/O transfers continues until the processing requirements of the transaction are satisfied. The transaction then terminates and leaves the system via the “COMPLETED TRANSACTIONS” arrow.

BEST/1 enables the analyst to represent cases where several transactions are in main memory and active at the same time.
time. One transaction may be using the CPU while another is using an I/O device. Similarly, several I/O devices may be providing service to several different transactions at the same time.

This overlap of processing activity—which is generated by several transactions that are sharing the main memory of a single computer system—is generally referred to as multiprogramming. Multiprogramming tends to raise the utilization of system resources and thereby improve certain measures of overall performance. However, the queuing delays associated with multiprogramming can have an adverse effect on other measures of system performance. BEST/1 has the ability to explicitly represent the multiprogramming activity described above, and the analyst can use BEST/1 to determine the effectiveness of multiprogramming in various applications.

THE SERVER CONCEPT

As discussed in the preceding section, a job or transaction which has arrived at a system and has been loaded into main memory spends its time alternating between the following two states:

- Using active processing resources such as CPU's and I/O devices
- Waiting to use such resources

In the context of BEST/1, active processing resources are referred to as servers. That is, servers are points within a system where queues can form and where processing can be carried out. Each circle in Figure 6 corresponds to a different server.

One of the most important aspects of Figure 6 is the assumption that each server in the network is capable of independent operation. That is, it is assumed that the CPU can be processing one request at the same time that an I/O server is processing another request. Likewise, it is assumed that two or more I/O servers can be processing requests at the same time.

The assumption of simultaneous processing capability plays a critical role when the real I/O devices in a computer system are mapped into BEST/1 servers. For example, in the case of a string of high speed magnetic tape drives connected to a single controller and channel, it is only possible for one device to be actively transferring data at any time. Hence the entire string of tape drives should be represented as a single I/O server in Figure 6. On the other hand, if each tape drive were connected to a dedicated channel and controller, simultaneous operation would be possible and each device would be represented as a separate I/O server.

The issues that arise when representing disks and drums are somewhat more complex. In virtually all modern disk drives and controllers, seek overlap is possible. This means that any number of disks can be carrying out independent seek operations at the same time. Simultaneous overlap of the rotational latency time is also possible for disks and drums which employ rotational position sensing (RPS). However, the number of actual data transfers which can take place at the same time is limited to the number of parallel I/O channels that are connected to a particular controller and string of devices.

In cases such as these it is preferable to represent each disk and drum as a separate I/O server within BEST/1. That is, each disk and drum is assumed to be capable of independent simultaneous operation. The service time at each device must then be adjusted to account for the additional delays that arise because of channel contention. For example, disk service time—which is normally the sum of seek, rotational latency, and data transfer time—must be increased so that it becomes the sum of seek, rotational latency, data transfer, and channel queueing time. The increased service time represents a real effect since measurement devices would actually record the disk as being busy during channel queueing time.

BEST/1 provides facilities for calculating the amount of channel queueing that will arise within a particular configuration, and for automatically increasing I/O service times accordingly. Thus, channel contention is accounted for even though channels are not explicitly shown in Figure 6.

THE DOMAIN CONCEPT

As illustrated in Figure 7 the three principal components of a BEST/1 model are a set of workloads, a set of servers, and a set of memory domains. Essentially, memory domains are mechanisms for controlling the sharing of memory among various workloads. All multiple workload systems incorporate some mechanism of this type to prevent one or two workloads from monopolizing main memory in an uncontrolled manner. For example, time sharing transactions may be assigned to a domain with a maximum multiprogramming level of 5 while batch jobs are assigned to a domain with a maximum multiprogramming level of 7. This means that no more than 5 time sharing transactions and 7 batch jobs would be allowed to enter main memory at a time.
single time. Terms such as "partition", "region" and "initiator" are used in different operating systems to refer to the concepts of multiprogramming level and domain.

The specific mechanisms used to implement the memory domain concept vary somewhat from one system to another. BEST/1 allows a wide range of mechanisms to be modeled. Essentially, the analyst can characterize domains in terms of their maximum multiprogramming levels, their average multiprogramming levels, or in terms of certain distributions.

Note that memory domains are closely associated with workloads. In particular, each workload must be assigned to some domain. BEST/1 permits several workloads to be assigned to the same domain; alternatively, each workload may be assigned to a separate domain if desired.

As a final point, it should be clear from the preceding discussion that the memory queue shown in Figure 6 actually represents a set of queues, one corresponding to each domain. BEST/1 treats the queues separately, and values such as queue length and waiting time are computed individually for each memory queue.

BEST/1 INPUT PARAMETERS

Figure 8 presents an example of all the input data needed by BEST/1 for a typical application. In this case, the system being evaluated is processing three workloads: Workload 1 represents a stream of batch processing jobs, Workload 2 represents a transaction processing application, and Workload 3 represents the load generated by a set of time sharing users.

The BEST/1 input parameters used to characterize this application are grouped together on a "per workload" basis. For each workload, there are a set of workload descriptors and a set of service requirements. These two sets of parameters are discussed below.

WORKLOAD DESCRIPTORS

The workload descriptors in Figure 8 are used to specify the following information:

- The workload type: batch processing (BP), time sharing (TS), or transaction processing (TP)
- A label used to identify the workload on output reports
- Information about the multiprogramming level of the domain which is associated with the workload
- For time sharing and transaction processing workloads, information about the external factors which generate arriving transactions

In all cases, the first workload descriptor in the formatted BEST/1 input file contains the workload type and the workload label. The second descriptor characterizes the multiprogramming level of the workload's domain. For Workload 1, the average multiprogramming level attained during times when there is a backlog in the input queue is specified as
3.8. Workloads 2 and 3 are assigned maximum multiprogramming levels of 4 and 6 respectively.

Additional descriptors are used in Workloads 2 and 3 to characterize the external factors which generate arriving transactions. For Workload 2, these factors are characterized by specifying an average arrival rate of 4000 transactions per hour. For Workload 3, the external factors are specified as 30 terminals with average think times of 25 seconds.

SERVICE REQUIREMENTS

The second set of input parameters that are provided for each workload are referred to as the workload's service requirements. Essentially, these parameters specify the total amount of service time that is required, per transaction, at each server in the system. For example, each Workload 1 transaction (i.e., each batch job) requires an average of 13000 msec of processing at Server 1 (the CPU), 1099 msec of processing at Server 2 (Drum 1), 1940 msec of processing at Server 3 (Drum 2), and so on. The service requirements per transaction are specified separately for each workload as indicated in Figure 8.

Note that the service requirements in Figure 8 refer only to the time that a transaction spends actually using a server. The time spent waiting in a queue before reaching the server is not included in the input data, but is instead computed by BEST/1 and used to generate output reports.

EXTENDED EXAMPLE

BEST/1 enables the analyst to extend the basic system description of Figure 8 in a number of ways. These extensions are introduced by associating additional descriptors with each workload, and also by adding certain "special server parameters" to the end of the input file. Figure 9 provides an indication of the range of extensions which are available to the analyst.

--- WORKLOAD 1 --- DESCRIPTORS ---
  BP  BATCH PROCESSING
  3.8  AVERAGE MPL
  3.0  PRIORITY
--- WORKLOAD 2 --- DESCRIPTORS ---
  TP  DATA BASE INQUIRY
  3200.0  TRANSACTIONS/HR
  1.0  PRIORITY
  1.0  DOMAIN NUMBER
--- WORKLOAD 3 --- DESCRIPTORS ---
  TS  TIME SHARING USERS
  6.0  MAXIMUM MPL
  30.0  NO. OF TERMINALS
  25.0  THINK TIME (SECS)
  2.0  PRIORITY
--- WORKLOAD 4 --- DESCRIPTORS ---
  TP  DATA BASE UPDATE
  800.0  TRANSACTIONS/HR
  1.0  PRIORITY
  1.0  DOMAIN NUMBER

<table>
<thead>
<tr>
<th>SERVER</th>
<th>WORKLOAD 1</th>
<th>WORKLOAD 2</th>
<th>WORKLOAD 3</th>
<th>WORKLOAD 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CPU-370/168</td>
<td>13000.0</td>
<td>150.0</td>
<td>320.0</td>
<td>425.0</td>
</tr>
<tr>
<td>2 DRUM 1</td>
<td>1099.0</td>
<td>75.0</td>
<td>92.0</td>
<td>115.0</td>
</tr>
<tr>
<td>3 DRUM 2</td>
<td>1940.0</td>
<td>125.0</td>
<td>175.0</td>
<td>235.0</td>
</tr>
<tr>
<td>4 TAPE</td>
<td>395.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5 TAPE</td>
<td>496.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6 2314</td>
<td>747.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7 2314</td>
<td>204.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8 3330 SYSTEM PACK</td>
<td>4642.0</td>
<td>13.0</td>
<td>80.0</td>
<td>13.0</td>
</tr>
<tr>
<td>9 3330/ASPGQ</td>
<td>12461.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10 3330/ASPG</td>
<td>12462.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>11 3330 SCRATCH</td>
<td>2474.0</td>
<td>20.0</td>
<td>42.0</td>
<td>240.0</td>
</tr>
<tr>
<td>12 3330 SCRATCH</td>
<td>2474.0</td>
<td>20.0</td>
<td>46.0</td>
<td>230.0</td>
</tr>
<tr>
<td>13 3330 SWAP AND USER</td>
<td>3034.0</td>
<td>100.0</td>
<td>100.0</td>
<td>350.0</td>
</tr>
<tr>
<td>14 3330 SWAP AND USER</td>
<td>3035.0</td>
<td>100.0</td>
<td>100.0</td>
<td>350.0</td>
</tr>
<tr>
<td>15 3330 MODI USER PACK</td>
<td>1073.0</td>
<td>15.0</td>
<td>200.0</td>
<td>75.0</td>
</tr>
<tr>
<td>16 3330 MODI USER PACK</td>
<td>1073.0</td>
<td>15.0</td>
<td>0.0</td>
<td>85.0</td>
</tr>
<tr>
<td>17 3330 MODI USER PACK</td>
<td>1727.0</td>
<td>20.0</td>
<td>0.0</td>
<td>165.0</td>
</tr>
<tr>
<td>18 3330 MODI1 USER PACK</td>
<td>1727.0</td>
<td>21.0</td>
<td>0.0</td>
<td>176.0</td>
</tr>
</tbody>
</table>

--- DOMAIN 1 --- DESCRIPTORS ---
  LABEL  DATA BASE TRANS
  4.0  MAXIMUM MPL
--- SERVER 1 --- DESCRIPTORS ---
  LABEL  CPU-370/168
  2.0  MULTIPLE SERVER

Figure 9—Extended input file
WORKLOAD DIFFERENTIATION

The most obvious difference between Figure 8 and Figure 9 is the fact that a fourth workload has been added. In this particular case, the additional workload is not intended to represent a new application being added to the system, although additional workloads could obviously be used for such purposes. Rather the new workload represents a more detailed model of the same system represented in Figure 8.

In the case of Figure 8 the transaction processing application is represented, using Workload 2, as a single input stream arriving at a rate of 4000 transactions per hour. BEST/l will treat this workload as a single entity and will report response time, queue length, and other performance values for the workload as a whole.

Suppose, however, that the analyst now wishes to explicitly represent the fact that 80 percent of the Workload 2 input stream corresponds to data base inquiry transactions, and 20 percent corresponds to data base update transactions. In other words, suppose the analyst wishes to obtain separate performance values for each of the two transaction types.

In this case the analyst can use two separate workloads to represent the transaction processing application. As illustrated in Figure 9, Workload 2 is now used to represent the data base inquiry transactions. The arrival rate is set to 3200 transactions per hour (80 percent of 4000), and the service requirements are explicitly shown. In addition, Workload 4 is now used to represent the data base update transactions. The arrival rate is 800 transactions, and a different set of service requirements are provided. BEST/l will then be able to compute response time, throughput, and other performance values for each transaction type.

LEVELS OF DETAIL

It should be emphasized at this point that the model represented by Figure 9 is not intrinsically "more correct" than the model represented by Figure 8. Both are "correct" in the sense that they both enable the analyst to investigate a certain set of questions concerning the performance of a real system. Since the model associated with Figure 9 is more detailed, a more detailed set of questions can be answered. However, the analyst is required to provide a more detailed set of input data in this case: specifically, the separate service requirements for each of the two transaction types.

Because of the ease with which the level of detail can be altered, BEST/l enables the analyst to select an initial level of detail, run through a number of cases, and then increase the level of detail if necessary for those cases which warrant further analysis.

The ability to adjust the level of detail in a BEST/l analysis is also important in applications involving capacity planning, design of new systems, and other situations where performance prediction is involved. In these cases the information available to the analyst tends to become more and more detailed as time progresses. BEST/l thus enables the analyst to use the most detailed model that is feasible at each stage, and to increase the detail in the model as more refined information becomes available.

SHARED DOMAINS

Another important extension of Figure 8 which is illustrated in Figure 9 is the concept of the shared domain. It is implicitly assumed that each workload in Figure 8 has a separate domain and a separate memory queue. However, this is not the case for Workloads 2 and 4 in Figure 9. Rather, these two workloads are assumed to compete for the single domain which was assigned to Workload 2 in Figure 8.

The concept of two workloads sharing a single domain is easy to represent in BEST/l. As illustrated in Figure 9, this is done by placing a domain identifier rather than a multiprogramming level in the descriptors of Workloads 2 and 4. A separate domain descriptor is then added to the end of the input file to indicate the multiprogramming level associated with that shared domain. This is shown in Figure 9.

Note that the analyst is not required to use a shared domain in this case. For example, Workload 2 could have been assigned a maximum multiprogramming level of 3, and Workload 4 could have been assigned a maximum multiprogramming level of 1. This corresponds to a situation where each workload has a separate domain. BEST/l will, of course, calculate a different set of performance values in this case. The analyst can then determine which memory allocation strategy is most appropriate in terms of the performance objectives of the original system.

PRIORITIES

BEST/l permits the analyst to associate a CPU dispatching priority with each workload. In the case of Figure 8 no priorities were specified so BEST/l assigned the same default priority to all workloads. Figure 9 illustrates the way priorities are added to the workload descriptors. As indicated in the figure, the transaction processing workloads have highest priority, the time sharing workload has intermediate priority, and the batch processing workload has lowest priority.

In cases where priorities are specified, BEST/l uses the conventional "preemptive-resume" dispatching algorithm found in most operating systems. In other words, a high priority request which arrives at the CPU interrupts and preempts any lower priority requests that may be present. Lower priority requests resume processing from the point of preemption after higher priority requests release the CPU.

MULTIPROCESSORS

BEST/l input files sometimes include "special server parameters" which refer to the servers within the system rather than individual workloads. The number of CPU's in a multiprocessor configuration is such a parameter. Figure
Principal Results Report
CPU and I/O Utilization Report
Response Time Profile
Average Queue Length Report
Average Number Waiting Report
Waiting Time Profile
Service Rate Degradation Report
Server Residency Profile
Queue Length Distribution Report
Concurrency Report
Memory Report
Throughput Report
Server Report
Waiting Time Percentages Report

Figure 10—BEST/1 output reports

Figure 11 illustrates the Principal Results report generated by BEST/1 from the input data shown in Figure 8. This report is organized by workload. For each workload, the report indicates the average response time per transactions, the average throughput in transactions per hour, and the CPU utilization associated with that workload. Total CPU utilization is also reported.

For example, in Figure 11 the batch processing throughput is 102 jobs per hour and the response times for transaction processing and time sharing are 3.17 seconds and 3.31 seconds respectively.

***PRINCIPAL RESULTS***

<table>
<thead>
<tr>
<th>WORKLOAD</th>
<th>RESPONSE TIME</th>
<th>THROUGHPUT</th>
<th>% CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 BATCH PROCESSING</td>
<td>133.98 SEC</td>
<td>102, PER HOUR 36.9%</td>
<td></td>
</tr>
<tr>
<td>2 DATA BASE TRANS</td>
<td>3.17 SEC</td>
<td>4000, PER HOUR 22.8%</td>
<td></td>
</tr>
<tr>
<td>3 TIME SHARING USERS</td>
<td>3.31 SEC</td>
<td>3815, PER HOUR 33.9%</td>
<td></td>
</tr>
<tr>
<td>TOTAL CPU UTILIZATION = 93.6%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 11—Principal results report

RESPONSE TIME PROFILE

Figure 12 illustrates the Response Time Profile report generated by BEST/1 from the input data shown in Figure 8. Essentially, this report gives the analyst a breakdown of the response time values that appear in the Principal Results report. That is, this report gives the response time, in milliseconds, at each server in the system. By adding up these individual response times, the total response time per transaction is obtained.

The Response Time Profile is particularly useful when it is necessary to determine why a workload’s response time value is so high, and how important each server and queue really is insofar as overall response time is concerned.

INTERACTIVE DIALOG

The primary mode of operation for BEST/1 is through interactive time sharing terminals. When operating in this mode, the analyst has access to a variety of functions. These functions are requested through an interactive dialog that has been extensively refined and enhanced during several years of actual use.

The complete interactive dialog is specified in the BEST/1 User’s Guide. Some typical functions which are easy to carry out using the dialog are listed below:

- Add servers or workloads
- Change server speed
- Change arrival rate, number of terminals, CPU dispatching priority, multiprogramming level, etc.
- Request one or more optional reports
- Save an input file for future use
- Direct output to any desired device
- Execute a pre-defined set of interactive commands stored in a file

It should be emphasized that this is only a partial list of the functionality available to the BEST/1 user.

In all cases, the interactive dialog has been structured using concepts that can be easily and directly understood by
the performance analyst. Knowledge of the underlying mathematical theory is not required to set up BEST/I models or exercise the interactive dialog.

**SUBROUTINE INTERFACE**

Users are sometimes interested in carrying out complex iterative analysis procedures which repeatedly invoke BEST/I. For example, a user may wish to increase a workload's arrival rate until the resulting response time reaches some threshold.

Some cases can, of course, be treated using the BEST/I interactive dialog. However, it is sometimes preferable to simply write a program which repeatedly calls BEST/I, analyzes the output, and adjusts certain BEST/I input parameters until a condition has been satisfied or a set of cases has been examined.

In order to facilitate this process, there is a special interface which allows BEST/I to be called as a subroutine by a FORTRAN program. The FORTRAN program can then be executed interactively or in batch processing mode.

**ACKNOWLEDGMENTS**

Special thanks are due to Professors Peter J. Denning and Stuart E. Madnick and to Dr. Gary Sockut for their helpful suggestions and comments regarding the design of BEST/I.

<table>
<thead>
<tr>
<th>RESPONSE TIME PROFILE BY WORKLOAD</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERVER</td>
<td>0 MEMORY</td>
<td>1 CFU—370/168</td>
<td>2 DRUM 2</td>
</tr>
<tr>
<td>0 MEMORY</td>
<td>0.0</td>
<td>838.1</td>
<td>221.5</td>
</tr>
<tr>
<td>1 CFU—370/168</td>
<td>65680.5</td>
<td>1197.9</td>
<td>1873.0</td>
</tr>
<tr>
<td>2 DRUM 2</td>
<td>1410.7</td>
<td>107.0</td>
<td>119.4</td>
</tr>
<tr>
<td>3 DRUM 2</td>
<td>3244.4</td>
<td>242.0</td>
<td>292.4</td>
</tr>
<tr>
<td>4 TAPE</td>
<td>397.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5 TAPE</td>
<td>500.2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6 2314</td>
<td>757.8</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7 2314</td>
<td>204.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8 3330 SYSTEM PACK</td>
<td>5865.2</td>
<td>17.1</td>
<td>105.6</td>
</tr>
<tr>
<td>9 3330/AS PQ</td>
<td>17053.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10 3330/AS PQ</td>
<td>17054.9</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>11 3330 SCRATCH</td>
<td>3005.0</td>
<td>79.5</td>
<td>523.0</td>
</tr>
<tr>
<td>12 3330 SCRATCH</td>
<td>3012.5</td>
<td>77.2</td>
<td>57.4</td>
</tr>
<tr>
<td>13 3330 SWAP AND USER</td>
<td>4675.4</td>
<td>231.8</td>
<td>156.4</td>
</tr>
<tr>
<td>14 3330 SWAP AND USER</td>
<td>4677.1</td>
<td>231.8</td>
<td>156.4</td>
</tr>
<tr>
<td>15 3330 MOD11 USER PACK</td>
<td>1477.6</td>
<td>37.3</td>
<td>277.6</td>
</tr>
<tr>
<td>16 3330 MOD11 USER PACK</td>
<td>1137.1</td>
<td>31.4</td>
<td>0.0</td>
</tr>
<tr>
<td>17 3330 MOD11 USER PACK</td>
<td>1909.1</td>
<td>55.4</td>
<td>0.0</td>
</tr>
<tr>
<td>18 3330 MOD11 USER PACK</td>
<td>1916.5</td>
<td>59.0</td>
<td>0.0</td>
</tr>
<tr>
<td>TOTAL RESPONSE TIME</td>
<td>133978.6</td>
<td>3167.1</td>
<td>3312.1</td>
</tr>
</tbody>
</table>

Figure 12—Response time profile

**BIBLIOGRAPHY**
