Discussed is an approach to evaluating and comparing system costs and benefits (value) to the user and to his employer in a scientific environment.

Necessarily a semiquantitative measure, value to the user implies a departure from usual system efficiency measures such as system throughput.

Evaluated are usage policies based on single-stream and dual-stream batch systems, and terminal-oriented time-sharing systems.

Cost-benefit evaluation of scientific computing services

by D. N. Streeter

Computing services and their management have developed to a state that now demands better tools of economic appraisal. Underlying this need is the increasing range of available system options, i.e., the variety of computing services from which users can choose to solve their problems, and the diversity of facilities among which the operations manager can choose for the services he provides.[1]

In the field of scientific computing, there has been an apparent tendency to neglect economic considerations, and—more generally—service-related aspects of operations management. This tendency may derive from the fact that many computing center managers have a scientific or technical background, and, therefore, do not feel comfortable using the subjective data of value estimates used in this paper. An example of this tendency is seen in the formulation of computing service objectives, which are usually stated in terms of system characteristics (such as throughput or turnaround time), rather than the value of service delivered.[2]

Based on a chapter of a book that is to be published by John Wiley and Sons, Inc., New York, New York, this paper describes experience gained over the past several years in providing a variety of computing services to staff members at the IBM Thomas J. Watson Research Center. It reviews attempts to understand and quantify the effects of various services on the

researcher and his work. The author's intention is to stimulate interest and debate on the inherent concepts, rather than to claim hard, transferable results.

The following information is provided to give an insight into the magnitude of computer usage at the Research Center and forms the basis for this study. (These data do not represent the total usage, since a heavy external teleprocessing load and administrative services are not included.) The computing center consists of the following primary systems:

- System/360 Model 67 Simplex with TSS/360
- System/360 Model 67 Half Duplex with CP-67/CMS
- System/360 Model 91 with OS/360 and including LASP, RJE, and APL/360

There is a total of about seven hundred individual users, the majority of whom are professional scientists. Of these, about seventy-five are classified as heavy users, two hundred twenty-five are moderate users, and the balance may be considered light users. All personnel in the study work in a single building where no one is more than three-hundred yards from the computing center. All terminal-oriented teleprocessing service uses the dial-up system of telephone extensions.

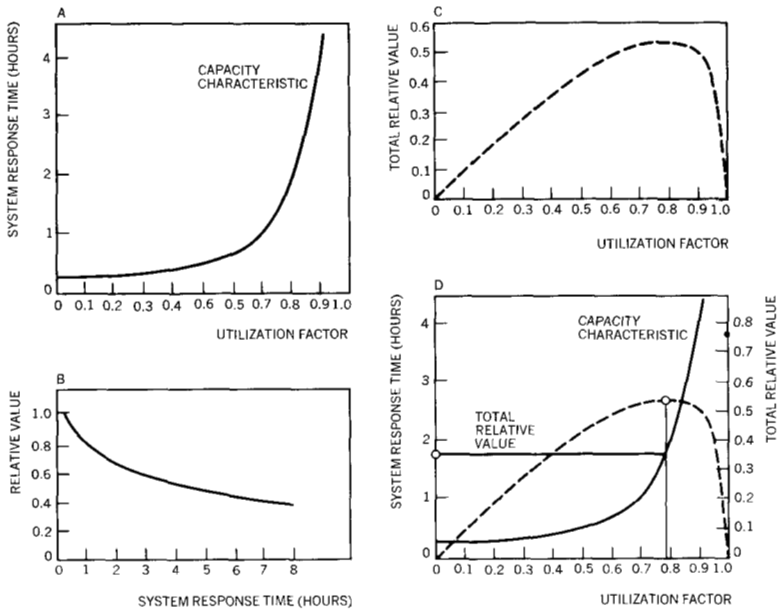
A single-job-stream batch processing system model illustrates the increasing system response time with system utilization. This is termed the system capacity characteristic. Also introduced is the observation that relative value of a computation to an experimenter decreases with time. Maximum relative value is obtained for this model. A two-job-stream model illustrates the effect of priorities both on value to the user and on system capacity. Using a test program, these factors are evaluated for a variety of systems, including time-shared systems. Emphasizing time-shared systems, the author evaluates system cost and user benefit tradeoffs for three modes of usage. By way of a method of differential costs and benefits in a time-shared environment, productivity estimates are made. Also compared are the relative values of a spectrum of services that a large research establishment might offer.

A conceptual model

one job stream

Arguments supporting the choice of net value of a service as the operational objective function to be maximized, for example, are discussed by Sharpe.[3] Based on this analysis, a conceptual model for visualizing batch operations is shown in Figures 1 and 2. In Figure 1A, the curve represents the *capacity characteristic* of a system, which shows the effect of system utilization on system

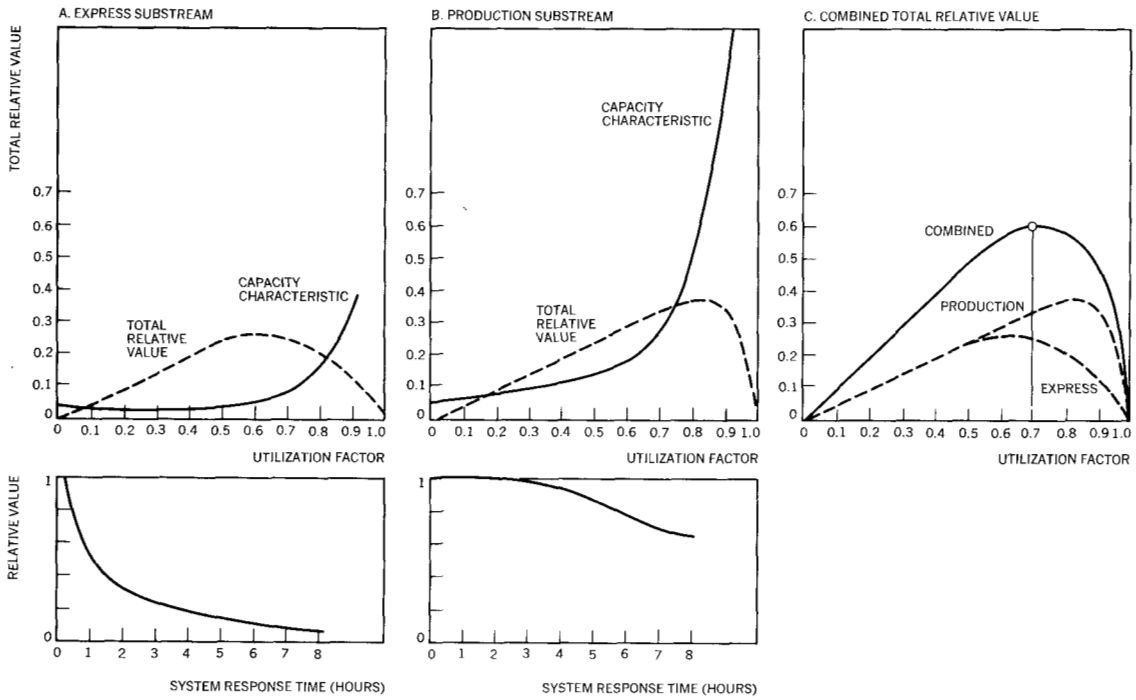
Figure 1 Single-stream batch processing



response time. The capacity characteristic curve here relates the *utilization factor* (the ratio of actual throughput to system throughput capacity) to system response or job turnaround time. The form of this curve can be derived from queuing theoretic considerations discussed in Reference 3, but it is more often determined empirically from observations of turnaround time under different levels of loading. The first model to be discussed is a single-stream batch processing system. Other models that use this characteristic are discussed later in this paper.

The curve shown in Figure 1B represents the averaged quantification of the subjective judgments of the diminution of the *relative value* of a job to the user as a function of system response time (with a fifteen minute response taken equal to unity). This curve has been determined primarily from responses to interviews or questionnaires submitted to users. Because the results have been independently checked in several ways, assume for now that such a curve can be determined with reasonable accuracy. Using the relative value curve in Figure 1B and the utilization characteristic curve in Figure 1A, the *total relative value* can be directly determined as shown in Figure 1C. Then, by overlaying Figures 1A and 1C, one can see in Figure 1D that operating the system at about eighty percent utilization maximizes the value of this system to this user population, assuming all jobs are given equal priority, in a single job stream. Correspondingly, we have observed that maximum value occurs when system response is just under two hours.

Figure 2 Dual-stream batch processing



two job streams

Consider now the operational alternative of dividing the job stream into two substreams, based on stated job urgency, and giving priority to one substream, called the *express substream.* The rationale for doing this derives from user interviews, wherein respondents who experience difficulty articulating the single relative-value curve in Figure 1B usually agree to the dual curves —*production* and *express*—as shown by the relative values of turnaround time in the lower boxes of Figures 2A and 2B.

When a user's problem requires the output of one job before he can enter the next job, speedy results are of greatest value to him. Such *sequentially organized* jobs benefit most from the short turnaround time of the express substream. This sequential mode is usually required during exploratory phases of research and during program development or debugging. At such times, the user's research progress relates directly to the number of runs he makes per day. Thus the total relative value curve for the express mode approximates a hyperbola, as shown in Figure 2A.

The complementary mode or substream is referred to as the *production substream.* Here one's program is essentially stable but requires execution with new data or parameter settings and generally involves less need for fast turnaround. One run a day is often sufficient, and, if more results are needed, several copies of the program can be run in parallel with different data adjoined.

The relative-value curve for the production substream, shown in the lower box of Figure 2B, tends to be flatter than that for the express substream. Although the instantaneous form of such a curve may change with time of day depending on the user's work habits, these relative-value curves have been averaged over time of day as well as user population. Capacity characteristics are shown in Figures 2A and 2B without ordinate scales simply for the purpose of comparison. For simplicity, the substreams are assumed equal in size. Therefore, the two capacity characteristics in Figures 2A and 2B average to the capacity characteristic in Figure 1A. Similarly, the relative-value curves in Figure 2 average to the relative value in Figure 1B.

Figures 2A and 2B effectively show batch-processing subtotals for the two substreams. The express total relative value is maximized by running with the system lightly loaded to keep the turnaround time short. Production total relative value on the other hand, is maximized by running the system at a high utilization. The value of the total work load, thus partitioned, realizes maximum total relative value at about seventy percent utilization as shown in Figure 2C. The main point of interest demonstrated by this example is that a two-stream system, running at about ten percent lower utilization than in the single-stream case, delivers about ten percent greater value to the user population.

Several comments are perhaps called for in connection with the preceeding and subsequent analyses. It is necessary to establish communication between operations management and the user community so that meanings and measures of "values" and "benefits" can be ascertained. Communication channels that have been employed include questionnaires and departmental user representatives. Perhaps more effective soundings could be taken during frequent conversations with a carefully selected small panel of users. Users and operations people should appreciate the necessity for articulating their thoughts and basing analyses and decisions on these subjective data. The nature and limited precision of these data should be kept in mind so that results of analyses are not exaggerated or misused. Concerning this last point, one should bear in mind that the examples given in this paper are drawn from our experience in a research environment and require the special assumptions used in this paper.

Three classes of interactive service

The foregoing discussion dealt with batch processing services. We now define and include the following three classes of interactive service: (1) computation only; (2) programming, debugging, and computation; and (3) problem formulation, programming, debugging, and computation. These definitions are intended to

clarify and add structure to the succeeding analyses of the costs of various services.

Class 1: computation only

The computation-only class is the most straightforward, and it is included in the succeeding two classes. Nevertheless, there is considerable difficulty in defining a cost function that includes all significant costs associated with the solving of a computation problem. In this definition, we attempt to assess these costs from the point of view of the management of the organization that employs the users and rents the computing and communications equipment. Thus it is reasonable to include in the cost function human and other costs as well as that of equipment costs. Also included in the costs should be a charge for time the scientist personally spends in solving a problem, charges for auxiliary services (keypunching, for example), and a delay penalty charge. A total cost formula that includes these charges may be expressed as follows:

$$T = S + C + U + D + A$$

where

T = total problem-solving cost
S = computing system costs (CPU time × CPU charge rate)
C = communications costs (connect time × communication charge rate)
U = cost of user's time
D = cost of delay (elapsed time × delay penalty charge)
A = auxiliary charges

Results of applying this cost formula to a specific problem are shown in Table 1. The problem used—ETEST—is a FORTRAN IV program that calculates the value of the base of natural logarithms e to 2500 decimal places. This problem places a fairly heavy computation load on the tested system while remaining relatively insensitive to human-response factors. Briefly, the program loops through 25 thousand iterations per line of output (or 2.5 million iterations for the complete problem). No penalty is imposed on machines without floating-point hardware since the entire computation is done in integer arithmetic. For the same reason, the problem is small enough to run on nearly any machine, while at the same time imposing a fairly realistic computational burden on it. These limitations often create some bias against the cost-effectiveness of larger machines, which have more comprehensive facilities. This bias, however, does not seriously affect our present use.

ETEST has been used informally in the industry as a basis for comparing the cost of using various systems. The extension here is to consider the other costs—in addition to the computer and com-

Table 1 Options and costs for the ETEST program

	System 360/91/OS	*System 360/67/TSS*	*System 360/67/ CP/CMS*	*IBM 1130*	*System/360 67/191 TSS/OS*	*System 360/67 Batch*
CPU time @ $/min. = system charge	0.25 min. @ $15/min. 3.75	2.56 min. @ $10/min. 25.60	2.26 min. @ $10/min. 22.60	21 min. @ $0.40/min. 8.40	0.25 min. @ $15/min. 3.75	0.71 min. @ $10/min. 7.10
Connect time @ $.06/min. = connect charge	–	22 min. 1.32	13 min. 0.78	–	10 min. 0.60	–
User's time @ $.50/min. = user time cost	21 min. 10.50	10 min. 5.00	8 min. 4.00	35 min. 17.50	16 min. 8.00	21 min. 10.50
Elapsed time @ $.10/min = delay cost	120 min. 12.00	22 min. 2.20	13 min. 1.30	155 min. 15.50	85 min. 8.50	120 min. 12.00
Auxillary charges	(Keypunching) 2.00	–	–	–	–	(Keypunching) 2.00
Total cost	$28.25	$34.12	$28.68	$41.40	$20.85	$31.60

*Assumes 2 hour wait for 1130 availability.

munication system costs – when running the problem on several systems. The source program consists of twenty-five FORTRAN statements that are entered via cards on batch systems and via IBM 2741 terminals on time-sharing systems.

Some comments on the charge rates are in order. System charges are based on the rule of thumb of five or six dollars per hour for each one thousand dollars of monthly rental.[4] The $30 per hour user time cost is calculated with an unusually heavy overhead burden placed on the scientist, first, because he is the sole source of output from the laboratory. Also, the Corporation obtains only value received from its Research Division, i.e., the division is not defined as a profit center. Therefore, the nominal utility of a scientist equals the budget divided by the number of professional staff members.

The delay penalty rate is elicited from user-panel estimates of the average lowering of productivity that results from not having the solution to the problem of greatest current importance. On the average, this is estimated to be about a twenty-percent reduction in productivity.

We may now make a few observations on the results shown in Table 1. First, the total cost of solving the ETEST Problem is roughly the same whether using the System/360 Model 91 with the System/360 Operating System (OS/360) or the System/360

Model 67 with the Time Sharing System (TSS/360) or the Model 67 with CP-67/CMS time sharing facilities. The fraction of the totals that are used for system communication charges vary widely, however. They range from 13 percent for the Model 91 to 79 percent for CP-67/CMS terminal I/O. The batch user time consists of three walks to the computing center, and the elapsed time includes keypunching and batch turnaround time. The IBM 1130 CPU cost is in the medium range for the group. Note, however, the total cost includes thirty-five minutes of user time to punch the cards and compile and execute the program. An important factor in this case is the delay cost, which can become high where such a hands-on machine is used extensively.

The most economical service for the job is the coupled TSS/360–OS/360 computer network,[5] which allows for terminal input via the Model 67 and TSS/360, program and data transmission, and then execution on the System/360 Model 91 with OS/360. Here, programs are executed in the batch mode and results are transmitted back to the user's terminal via the Model 67. Such a hybrid operation acquires some of the advantages of both interactive and batch modes.

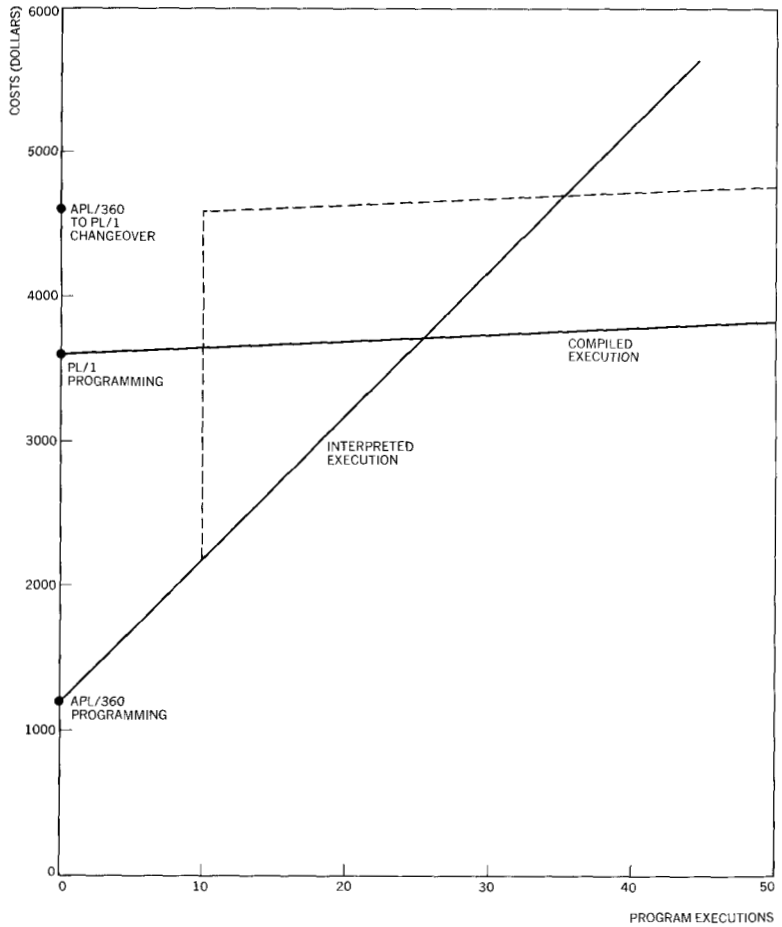
Class 2: programming, debugging, and computation

The analysis, when extended to include the programming and debugging functions, becomes more difficult, primarily because of greater variability in programming styles and proficiency. It is possible, however, to make a few observations about the choice of services at this level. The problem considered here is whether to use an interpretive or a compiling language processor on a given problem. The Research Center provides APL/360 as an interpretive facility, and FORTRAN and PL/1 as compiling systems.

On several problems that have been coded both in APL/360 and in either FORTRAN or PL/1, experience shows that it takes about three times as long to program and debug a problem using FORTRAN or PL/1 as it does using APL/360. On the other hand, the interpreted execution of the APL/360 program usually costs a factor of ten to a hundred more than the execution time to obtain a solution to the same problem using a compiled program. Figure 3 shows graphically the costs of programming plus execution for the two hypothetical programs, as a function of the number of executions. In this example, (based on a factor of three difference) it is assumed that the programming costs are $1,200 for APL/360 and $3,600 for PL/1. The execution cost is $5.00 per run for PL/1 and $100 per run for APL/360.

Using those assumptions, and neglecting factors other than programming and execution costs, the following conclusions emerge. The choice of a compiling or an interpreting processor should be based on the expected number of runs of the program. In a research laboratory, the evolving nature of many projects dooms

Figure 3 Hypothetical programming and execution cost comparison

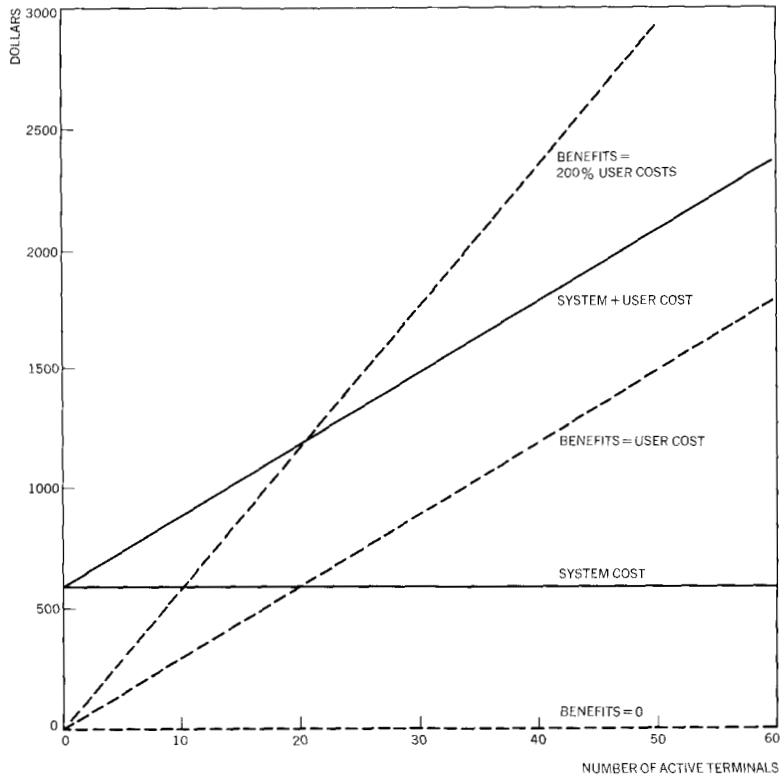


most programs to a relatively short life span. This perhaps favors the use of an interpretive processor more than other environments would. If, however, the program life expectancy is not known until after development begins and then appears lengthy, reprogramming should be encouraged and facilitated. The dashed line in Figure 3 shows such a path, assuming the PL/1 programming time is reduced by one-third as a consequence of the APL/360 experience.

Class 3: problem formulation, programming, debugging, and computation

We now come to the most comprehensive and difficult class of computer utilization for a research organization. This class involves the entire problem-solving process. Here we consider a means of quantifying benefits of using time-sharing services in the problem-solving process. Before presenting our findings at the Research Center, we summarize the results of an experiment conducted by Gold,[6] who compared problem-solving cost and

Figure 4 Benefits and costs in a time-shared system



effectiveness using time-sharing and batch-processing computer services. Gold's subjects were a class of sixty-six graduate students who were seeking ways of improving a business operation through a programmed simulation model of the business. Half of the students used time sharing and the other half used batch processing. All other factors, such as language skill and hardware, were assumed to be equal.

The salient findings of Gold's research are the following. Time-sharing users required five times as much computer time as the batch-processing group. The average man time required by the users of the time-shared computer system was 16.0 hours, whereas users of the batch system expended 19.2 hours. Therefore, at the given system charging rates, the total cost of solving the problem was less when using time sharing under the assumption that the user time is worth more than $11.83 per hour. An average of 6.5 hours of man time was expended before finding the most successful decision rule for the time-sharing user, as compared with 12.0 hours for the batch user. For the batch user, the average increase in simulated profit was $244, whereas, the corresponding increase was $444 for time-sharing users. This was

greater than an eighty percent improvement in performance. Gold's analysis "of the students' perceptiveness and understanding of the problem" showed significantly higher grades assigned to the users of the time-sharing system.

time-sharing cost-benefits

Gold's findings correlate strongly with experience in time-sharing usage at the Research Center. In Gold's study, the students' repeated solutions of the same problem made possible a controlled experiment. Such an opportunity for control does not arise in a research laboratory where each user works on a different problem.

Referring to Figure 4, it is feasible to analyze the time-sharing usage costs in a statistical sense. Here the utilization factor is expressed in terms of number of active terminals. The system cost (rental) is independent of utilization, as is the case for a single-function, in-house system, and is represented by the flat rental curve in Figure 4. The value of the users' time is a $30 per hour nominal utility figure. From the representation in Figure 4, we can identify the following three simple cases:

- For users who are doing no useful work (benefits equal zero) the system-plus-user cost curve represents the combined cost of system and user time, which merely increases with the number of users without any possibility of justification.
- For users who are doing only as much work as they would without the assistance of the time-sharing system, (benefits equal user costs) the user costs vanish—as shown by the system-plus-user cost curve—but leave the system rental as an unjustified cost.
- Users doing more work by using time sharing increase the total (system-user) utility, resulting from a positive increment in productivity, which becomes a benefit. If, for example, the average user's productivity is doubled, the benefits-equal-200% cost curve shown in Figure 4 represents the benefit realized.

Consider now the more complete representation shown in Figure 5, where the following refinements have been introduced. System costs are shown here as the differential cost between the time-sharing service and some alternative service to which it is compared (e.g., slide rules or batch computing). The differential system cost is assumed here to be $400 per hour. The cost-benefit curves are labeled with differential productivities that would be achieved on an instantaneously responsive system. That user productivity degrades as the system becomes more heavily loaded and slows down is shown by the saturation and decline of the cost-benefit curves. The magnitude of performance degradation takes into account the fact that user productivity is proportional to the number of interactions per unit time, where

Figure 5 Differential system costs and productivity benefits

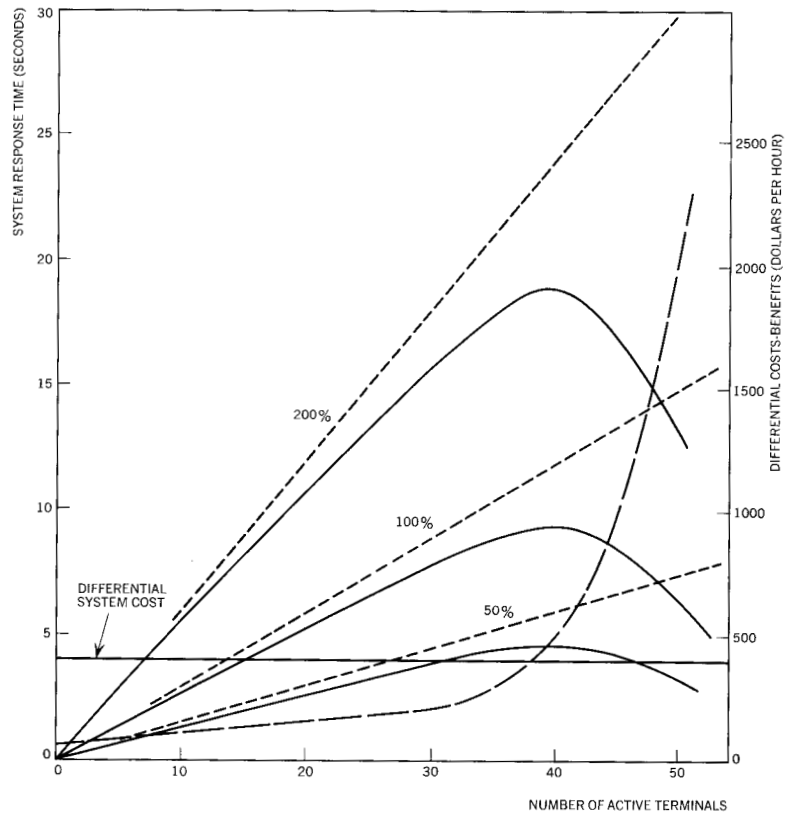
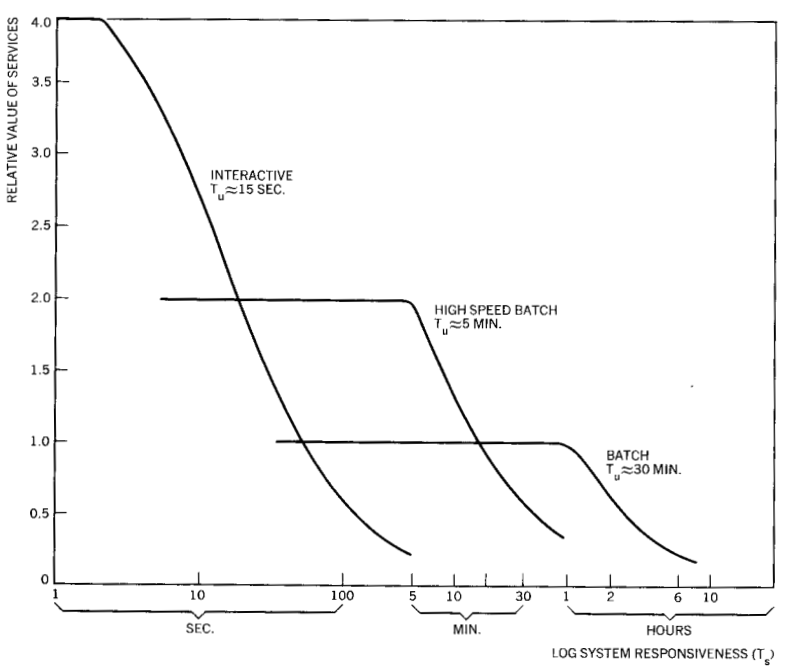


Figure 6 Spectrum of relative values of computing service



$$\text{Interactions per hour} = \frac{3600 \text{ (sec/hour)}}{I \text{ (sec/interaction)}} = \frac{3600}{S + U}$$

Here

S = mean system response time from Figure 5
U = mean user response time (assumed to be 15 seconds)

We conclude from this analysis that a time-sharing system is justified where the differential productivity benefits exceed the differential system cost for the average level of loading on that system. Note here that the family of percentage lines represent positive differential productivity based on instantaneous system responsiveness. Thus, in Figure 5, the system is barely profitable where the average differential productivity is 50%, and the average number of active users is between 32 and 46.

At the Research Center, we generally estimate the differential productivity of users whose work requires interaction to range between one hundred and three hundred percent. We attempt to encourage discriminating use of available services, both by allocation and by pricing policies. Also encouraged is the use of the computer network, which facilitates the shipment of jobs that do not require interaction (such as compilations, assemblies, executions) to the batch processor.

a spectrum of services

We now compare relative values of three services on a continuous spectrum of system responsiveness as illustrated in Figure 6. Batch service with turnaround time of one hour has a normalized value of one. The resubmission time for a batch job is assumed to be 30 minutes. The reduction in value of batch service with longer turnaround time is illustrated in Figure 6 by the falling off of the relative value curve as response time increases. The curves in Figure 6 are calculated by the following formula:

$$\text{Relative value} = \text{peak value} \times \frac{T_U + T_{SO}}{T_S + T_U} = \text{peak value}$$

$$\times \left(\frac{1 + T_{SO}/T_U}{1 + T_S/T_U} \right)$$

where T_{SO} is the nominal system responsiveness associated with the peak value (which is 60 minutes in the batch case).

Similar curves are shown for high-speed batch and interactive services. For high-speed batch, we assume that the user waits for his output, scans it quickly and resubmits his job in five minutes. The more responsive the system is, the more valuable is its service to users who require many turnarounds. Again, the value falls off sharply with increasing system response time T_S as

the user waits for his output. This sensitivity is reflected in the above formula by the critical ratio T_S/T_U. Similarly for interactive services, there is also a characteristic user response time (about 15 seconds), and a potentially high service value that degrades rapidly as users waste time at the terminal waiting for the system to respond.

To summarize these comparative cost-benefit considerations, there are several types of computing services that can be roughly characterized by a nominal system responsiveness or peak value of the service and by a characteristic user responsiveness in that mode. The value of each type of service quantitatively degrades as a function of the ratio of system responsiveness to user responsiveness.

Concluding remarks

Evaluation techniques for measuring computing system efficiency continue to be refined as the variety of systems and their applications expand. System measures often tacitly assume that the highest overall efficiency is achieved when a certain critical system parameter, such as throughput, is optimized. Recognizing that these measures may be quite valid when the computing system essentially models the business, this paper does not differ with such criteria. Used in a research environment for supporting individual research workers, however, system measures and policies should attempt to be more flexible, and should strongly consider the human costs and benefits involved. The reason for this is that research projects are so varied that a single system policy may not effectively model the "business" of research. Therefore, we attempt to evaluate and compare system costs and user benefits under a variety of user policies.

For batch systems, we have studied a two-substream policy—express and production—depending on the value of speedy results to the user. On a two-substream basis, we have found that there can be an increased relative value of the system output to users at an overall system utilization lower than that resulting from a single-stream policy.

In the case of interactive systems, we show the effects of several user policies. Here we conclude that such policies should encourage the use of interactive systems where the user benefits exceed the sum of all cost factors at least by an amount that the manager establishes as minimum.

The immediate advantage of policies that encourage the right choice of computing service is the more effective use of equipment. Still larger is the advantage of helping people become more

effective and productive. The importance of achieving such improvements has been pointed out by Peter Drucker,[6] who wrote: "Productivity will . . . be a major challenge and a major concern of the next ten years. . . . The 'cost-squeeze' of today, on governments, universities, and business, is the first warning—it is really a productivity squeeze. The bulk of tomorrow's employment will be in the service trades, knowledge jobs—in health care, teaching, government, management, research, and the like. And no one knows much about the productivity of knowledge work, let alone how to improve it."

ACKNOWLEDGMENTS

The author wishes to thank Professor Bernard Galler of the University of Michigan for providing the ETEST program, which is the work of Steven Lundstrom.

CITED REFERENCES

1. H. Sackman, *Man-Computer Problem Solving; Experimental Evaluation of Time Sharing and Batch Processing*, Auerbach, New York, New York (1970).
2. S. Stimler, *Real-Time Data Processing Systems*, McGraw-Hill Book Company, New York, New York (1969).
3. W. F. Sharpe, *The Economics of Computers*, Columbia University Press, New York, New York (1969).
4. W. P. Hegan, "Buying and Selling Computer Time," *Computers and Automation* **17**, No. 9, 32–40 (September 1968).
5. W. S. Hobgood, "Evaluation of an interactive-batch system network," *IBM Systems Journal* **11**, No. 1, 2–15, (1972).
6. M. M. Gold, "Time-sharing and Batch-processing: An Experimental Comparison of their Values in a Problem-solving Situation," *Communications of the ACM* **12**, No. 5, 249–259 (May 1969).
7. P. F. Drucker, "The Surprising Seventies," *Harper's Magazine*, 35–39 (July 1971).