A probability-based, theoretical model of a multiprogrammed computing system is suggested for planning future computing center requirements.

Validation of the planning model is attempted with respect to the theoretical model and applications to short-range and long-range planning.

Modeling for computing center planning

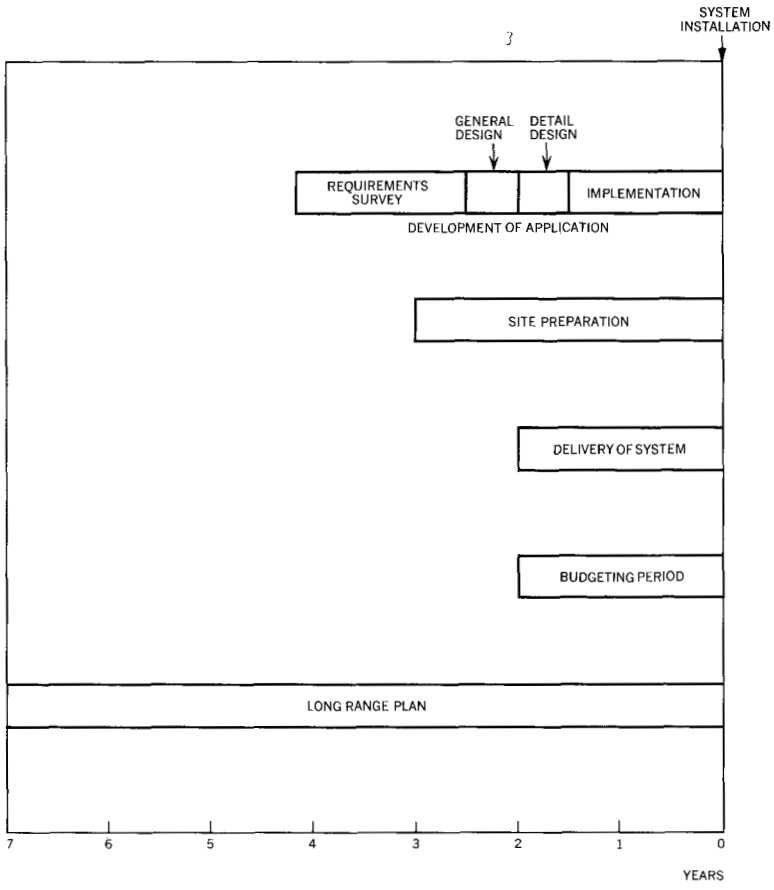
by F. Hanssmann, W. Kistler, and H. Schulz

a current forecasting technique

A computing center's current capacity is based on a prior forecast of what the center's work load is today. Thus in the planning phase, projected computing requirements are translated into time estimates for alternative system configuration under consideration. The time requirements may then be compared with the system's capacity in time units. Existing techniques for estimating time requirements are usually based on the notion of run times of individual jobs, which comprise the time interval from start to termination of each job. A uniprogramming system, for example, has sufficient capacity if the sum of run times for all jobs does not exceed a realistic operating time for the system during a given time period. In a multiprogramming environment, run times are assigned to individual regions, so that total capacity (region time) is multiplied accordingly. In principle, however, the technique is the same.

One technique currently used for estimating run times for jobs is as follows. Based on detailed jobs specifications, the input/output (I/O) times for several different categories of I/O devices are estimated. Central processing unit time is ignored as being irrelevant to the estimate. (Throughout this paper, the notions of I/O time and CPU time refer to purely productive times, when these devices are in operation, rather than to elapsed time for completion of jobs.) All I/O times are added on a summary sheet and multiplied by an empirical correction factor of <1, which

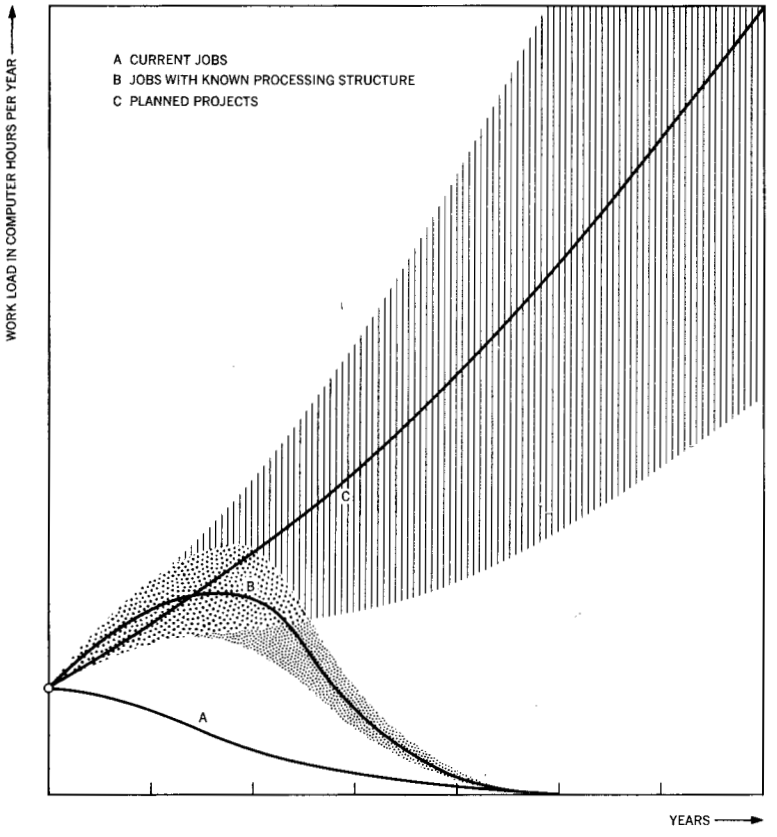
Figure 1 System planning horizons



reflects the extent of overlap among I/O processes. The result is the estimate of run time for each job. This technique may be generalized for the case of multiprogramming; it then requires different empirical correction factors.

Weaknesses in this planning technique are evident. Its basis is purely empirical and is, therefore, tied to an existing specific system structure. Thus the method fails even for such small configuration changes as tape speed or the number of regions, and reliable comparison of system alternatives becomes impossible. Another grave weakness is the impossibility of giving sufficient detailed specifications of jobs such as are required for estimating time requirements. Figure 1 illustrates planning horizons that a large computer installation has to observe to provide adequate facilities in a growing business. Ideally, the basic system structure should be specified three to four years before installation. Figure 2 indicates the growth in uncertainty of job composition as one projects current knowledge into the future. Experience teaches us that curves A and B illustrate the future effect

Figure 2 Forecast of work load composition



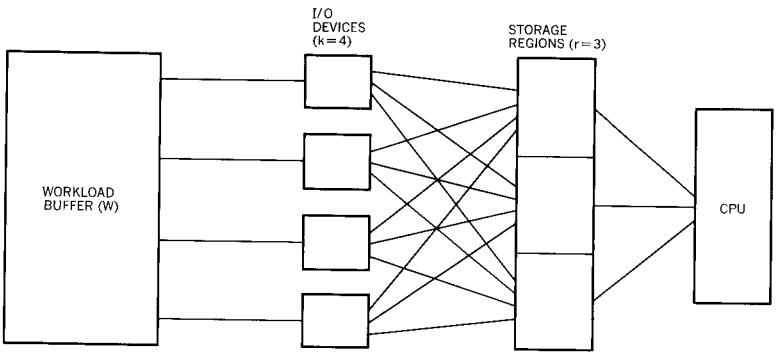
of estimated work loads based on job specifications whereas the bulk of the actual work load consists of presently unknown jobs (curve C).

a new planning model

The planning model discussed in this paper must of necessity be a relatively crude macromodel so that its predictive value will be useful approximately five years in advance (as shown by curves A, B, and C in Figure 2). The model does not require a detailed description of individual jobs, rather it uses a general description of work load. Furthermore, the model has a theoretical basis that gives it an explanatory nature that has general validity beyond a specific existing configuration. Of course, such a theoretical model must be tested within the range of experience before it can be accepted as the basis for planning.

A brief introduction to the theoretical model and its validation are necessary background for an understanding of the planning model. Use of the planning model for short- and long-range planning exemplify the broad scope of forecasting possibilities.

Figure 3 Structure of the Gaver model



Theoretical model

The macromodel we suggest for system planning is a modification of a model that has been developed by Gaver.[1] In this model, a computer system is deliberately simplified and conceptualized into the structural elements shown in Figure 3. In the course of our study, I/O devices are interpreted as channels (contrary to the ordinary meaning of the term I/O device). The work load is conceptualized as a set of "program segments," each of which must pass through three operations: input to one of the storage regions, processing by the CPU, and output. The model makes no distinction between different jobs or applications. The work load is characterized solely by the statistical properties of the segments, that is, their I/O times and CPU times.

The principal assumptions of the theoretical system model are:

- Jobs are processed by the system, segment by segment.
- Each segment requires one storage region and one I/O device.
- CPU time per segment is a random variable (α) with a stationary probability distribution, which we normally assume to be an exponential distribution.
- I/O time per segment (including I/O time between two consecutive processing times) is also assumed to be a random variable
- The work load buffer is infinite.
- Main storage size per storage region is constant. Thus, a change in the number of regions (r) implies a corresponding change in total storage size.

Two implications may be seen in these assumptions. Since a storage region uses one I/O device, no waiting for I/O operations occurs if the number (k) of channels equals or exceeds the number (r) of regions. In fact, there is no advantage for k ever to exceed r. Furthermore, the infinite work load buffer guarantees that a region is always occupied if an I/O device is available. These two properties of the theoretical system are only ap-

proached in the real system, and appropriate corrections must be applied when comparing the two systems.

mean CPU productivity

Consider first some relationships within the simplified theoretical system. By the preceding assumptions, the processes in the system are determined within the limits of stochastic variation. In fact, the processes could be simulated by taking samples of processing times and I/O times, thereby fully determining the *mean productivity* of the CPU. (This quantity plays a key role in our planning calculations.) A *busy period a* for the CPU terminates if the CPU is unable to find another region for which I/O operations have been completed. The busy period is followed by a *waiting period w*. Using these definitions, we now define the *mean productivity* $\bar{p}$ of the CPU as

$$\bar{p} + \frac{\bar{a}}{\bar{a} + \bar{w}}$$

Note that only mean values of the variables enter the mean productivity definition. Gaver computed mean productivity without recourse to simulation by the complex application of probability theory. He presented his results for different types of probability distributions of compute times and I/O times per segment.

mean computing intensity

If we designate the mean values of the CPU and I/O time per segment by random variables $\bar{\alpha}$ and $\overline{\beta}$, respectively, we may define *mean computing intensity* as the ratio of the CPU time per segment to the I/O time per segment

$$\overline{\lambda} = \frac{\bar{\alpha}}{\overline{\beta}}$$

Computing intensity is not a normalized ratio, and it may exceed the value of 1.0. Assuming exponential probability distributions, CPU productivity depends only on the configuration parameters r and k and the mean computing intensity as follows:

$$\bar{p} = g(\overline{\lambda} | r, k) \qquad (1)$$

mean productivity per region

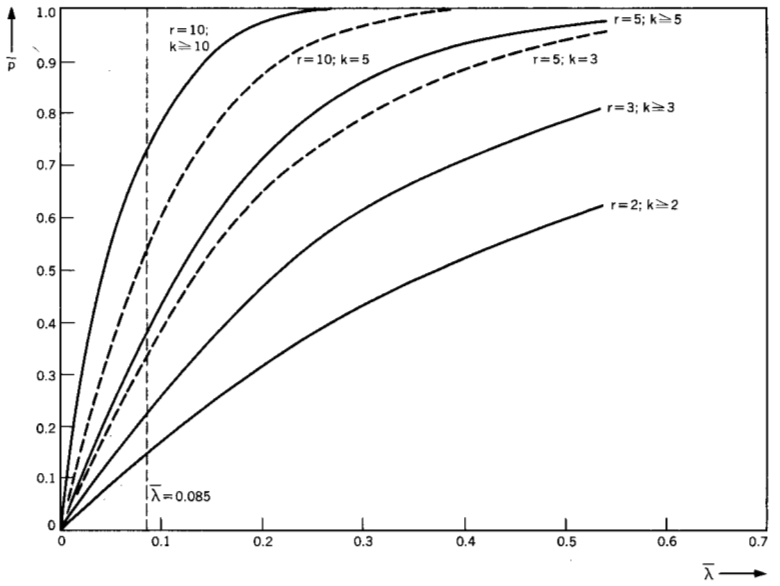
Gaver's research does not yield the function g in explicit form. We obtained it by recursive computation. For the case of exponential distributions, Figure 4 exhibits several productivity curves based on tables published by Gaver. Not surprisingly, productivity increases monotonically with computing intensity as well as with the number of storage regions. Given Equation 1, we may easily write the mean productivity per region

$$\bar{p}' = \frac{\bar{p}}{r}$$

as well as the mean channel productivity

$$\overline{q} = \frac{\bar{p}}{\overline{\lambda} k}$$

Figure 4 Theoretical productivity curves



corrected mean CPU productivity

Since in practice the work load buffer is not infinite, nonavailability of work causes idle time for the regions. (This idle time should not be confused with that caused by nonavailability of the CPU.) For this reason, the nominal number of regions r in the theoretical model should be replaced by the mean effective number of regions $\bar{r}_e$. Furthermore, realistic I/O processes may require several channels rather than one, and these may be dependent on each other or interfere with each other. If we wish to describe the much more complex reality by an equivalent single-channel model, we must increase the true I/O times by a suitable correction factor γ. Consequently, we must replace the mean computing intensity $\bar{\lambda}$ of the theoretical model by the effective computing intensity $\gamma\bar{\lambda}$. We thus obtain the following version of the corrected mean productivity model:

$$\bar{p} = g(\gamma\bar{\lambda}|\bar{r}_e, k) \tag{2}$$

Our planning technique is based upon the corrected mean CPU productivity model. For validation of this model, as well as estimation of the two correction factors, we must proceed empirically.

Model validation

Data for validation of the model were obtained by measurement and observation of the existing System/360 Model 65 for one 16-hour period. To obtain confidence in the model, it is necessary to observe wide excursions of the variables concerned. For this reason, we measured or estimated the computing intensity,

effective number of regions, and CPU productivity for relatively short time segments. The following direct measurements were made:

- Measurements of region occupancy based on start and stop times by program step
- Types of jobs
- CPU productivity (including system tasks) by time segments of 10^{-4} hours
- CPU time by program step

Additionally, we estimated or computed I/O times of the various categories of I/O devices (grouped by channel and based on data volume per step) since I/O related quantities were not directly measurable.

These measurements and estimates were used to perform three computations for time blocks of varying length:

- Allocated portions of CPU and I/O times by program segments (Inaccuracies were caused by the fact that program segments cross the boundaries of time blocks.)
- CPU time of system tasks by subtracting estimated CPU time of program segments from measured total CPU time
- I/O time of system tasks

Some results of the direct measurements of CPU productivity and storage-region occupancy are shown in Figure 5. Table 1 presents a preliminary averaging of the data in Figure 5. The model showed that CPU productivity was not very high. We also calculated an average storage region occupancy of 2.6 (out of four storage regions excluding storage occupied by system tasks). Since in our experimental configuration the number of channels exceeded four, the validity test of the model is, therefore, based on the assumption $k \geq r$.

Measurements for individual program segments (in the sense of the Gaver theory) could not be obtained because a program step is normally much longer than a segment. Therefore, in place of the CPU time per segment, we consider the CPU time per step divided by the number of I/O processes. This quantity x could be estimated from the histogram of the random variable x as a function of absolute frequency in Figure 6. Here we see that x is approximately exponentially distributed. Figure 7 exhibits the results of a similar test on a logarithmic scale. Comparing the straight line with the spread of experimental points, we conclude that the assumption of exponential distributions (straight line on the logarithmic scale) is approximately satisfied.

We are mainly interested in relationships among productivity, computing intensity, and effective number of regions. In order to generate a larger number of points for comparisons of empirical

Figure 5 Detail from direct measurements

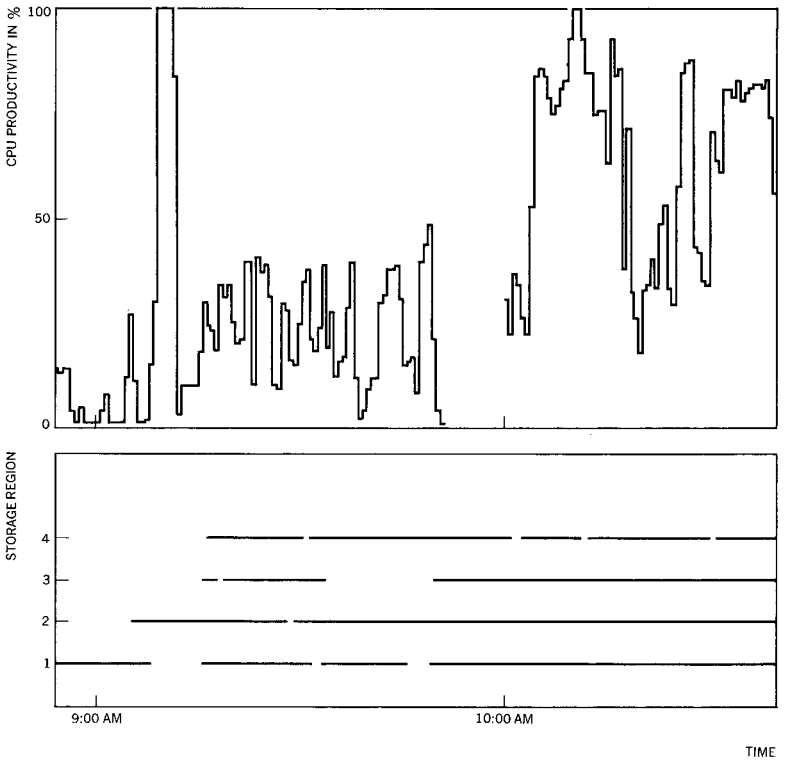


Table 1 Results of direct measurement

Direct measurement	*Time (hours)*
CPU active	3.3434
CPU wait	7.3366
Other	6.1846
Total	16.8646
Indirect measurement	*Computed results*
Productivity (of regions 1 to 4)	25.6%
Productivity (including system tasks)	31.3%
Average regions occupied (in a total of 4)	2.6%

and theoretical relationships, we start by dividing the measurement period into 30-minute blocks. Each block is further subdivided into "homogeneous" blocks. (We call a time block homogeneous if the number of occupied regions does not vary within the block.) Part of the list of homogeneous blocks is shown in Table 2. The number of occupied regions per block is exclusive

Figure 6 Histogram of CPU time intervals per step

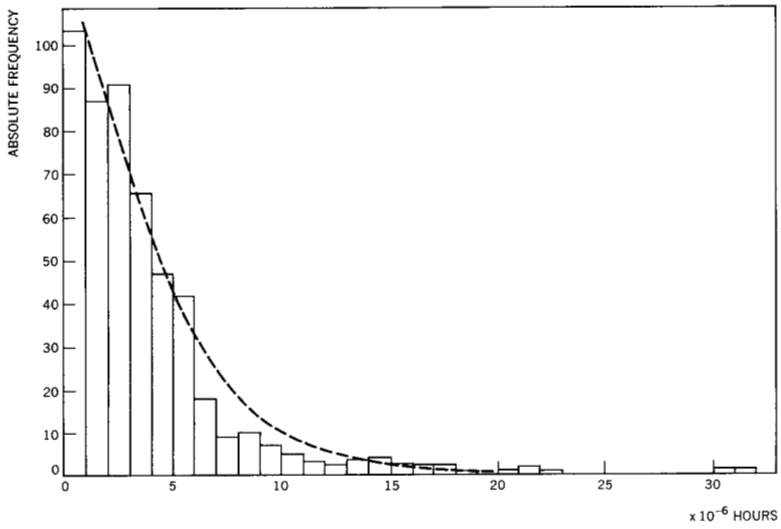


Figure 7 Test for exponential distribution

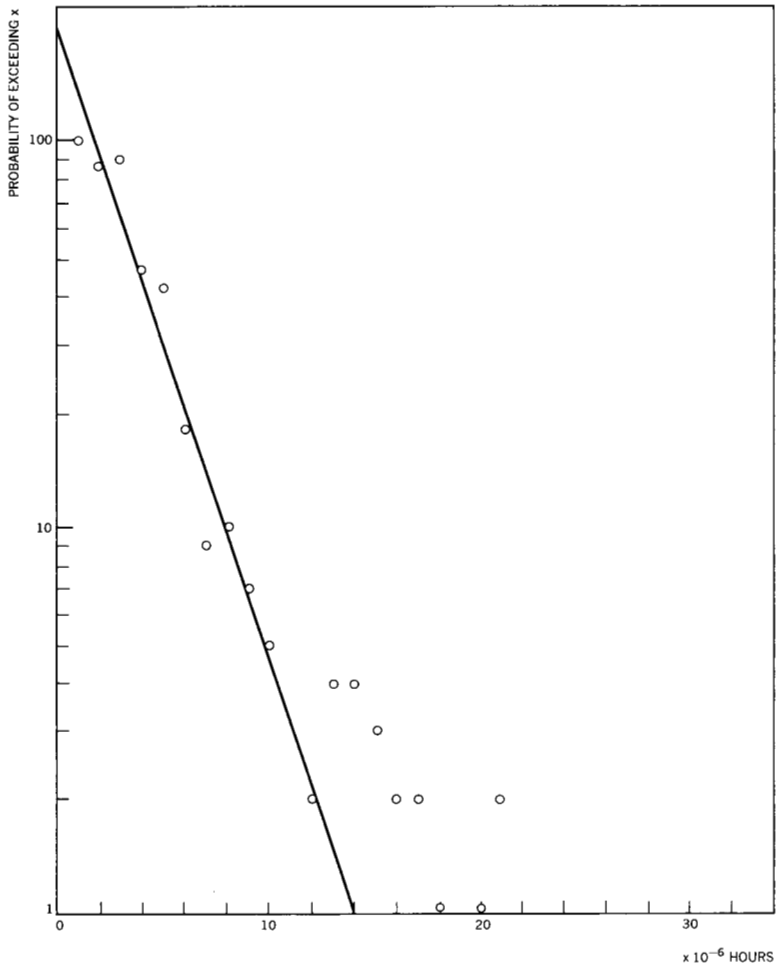


Table 2 Separation of data into homogeneous time blocks

Block number	r_e *(including system tasks*	*Block length* (10^{-4} *hours)*	*Total CPU time* (10^{-4} *hours)*	*Sum of I/O times* (10^{-4} *hours)*	λ *total*	*p total (percent)*
1	1	2325	94	259	0.363	3.9
2	2	705	49	39	1.256	7.0
3	3	3155	519	3055	0.170	16.7
4	2	210	38	167	0.228	19.0
5	1	279	38	14	2.714	12.7

Table 3 Median productivity and computing intensity for homogeneous groups

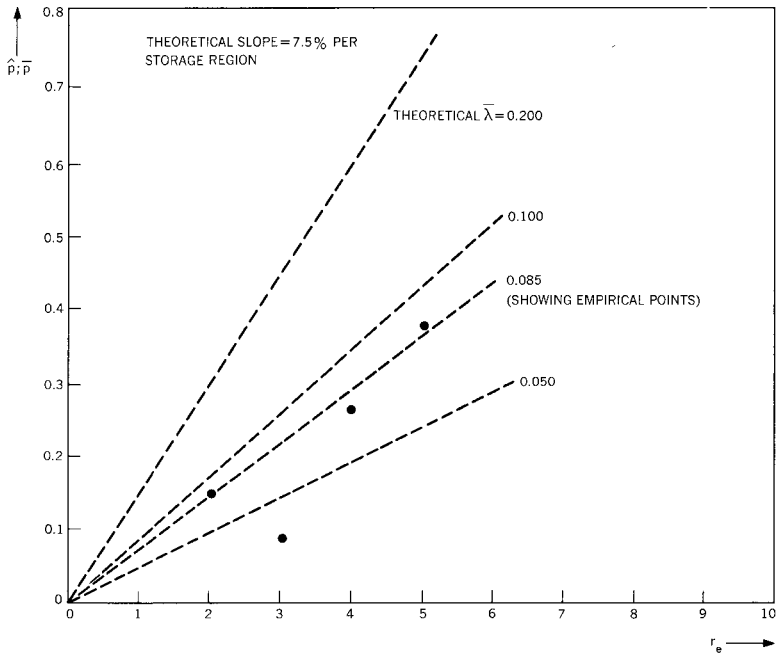
Group	r_e *(including system tasks)*	*Number of points*	$\hat{\lambda}$	$\hat{p}$	$\bar{p}$ *(for* $\bar{\lambda} = 0.085$*)*
1	2	2	0.31	0.15	0.15
2	3	7	0.34	0.08	0.22
3	4	12	0.59	0.27	0.30
4	5	15	0.34	0.38	0.39
5	aggregate	40	0.35	–	–
6	30-minute blocks	25	0.30	0.28	–

of system tasks. By contrast, all times are total times including those for system tasks. For this reason, we raise the number of regions by one before we proceed with testing the model. The simplest possibility of defining the computing intensity in multichannel system is to take the ratio of CPU time to the sum of all I/O times for all channels that are active during the time block under consideration.

correction factor for computing intensity

Homogeneous blocks are grouped together in Table 3. The validity test of the model must be made within these groups. For each group, the median values of productivity and computing intensity are noted. (Median values are used for simplifying the computations; exact computations of arithmetic means are equally possible.) Not surprisingly, median values of $\hat{\lambda}$ are highly compatible with a single true value of the mean computing intensity $\bar{\lambda}$. With one exception, the median values of productivity increase monotonically with the number of regions. As shown by the points in Figure 8, the relationship appears to be linear within the range of our measurements. This linearity is in agreement with the theoretical model, as is shown in Figure 8 by a family of theoretical curves for four values of $\bar{\lambda}$ from Figure 4. The regression line through the empirical points coincides roughly with the theoretical line for $\bar{\lambda} = 0.085$ shown by the last column of Table 3. Since

Figure 8 Theoretical and empirical productivities as a function of the number of regions



the empirical median of computing intensity is 0.35, we obtain the correction factor γ as follows:

$$\gamma = \bar{\lambda}/\hat{\lambda} = 0.085/0.35 = 0.243$$

productivity

This means that we have to make a correction of the actual input/output times by almost a factor of four to represent the empirical conditions by an equivalent one-channel model. We note that the slope of the empirical line indicates 7.5 percent additional productivity per region. We may now use the corrected theoretical model to estimate how the productivity curve levels off for higher numbers of regions.

In a similar manner, we can examine the dependence of productivity on computing intensity λ. The results of this test are shown in Figure 9 for homogeneous group number four of Table 3 (five regions, including system tasks). The theoretical curve is based on the previously determined correction factor γ. It coincides with the dispersion of the empirical results, especially in the median range of λ, which is of primary importance for planning purposes.

Finally, Figure 10 is devoted to the relationship between total productivity and productivity excluding system tasks. Theoretically, no point should lie above the line that passes through the origin with a slope of one. Occasional exceptions are explained

Figure 9 Productivity as a function of computing intensity for homogeneous group $r_e = 5$

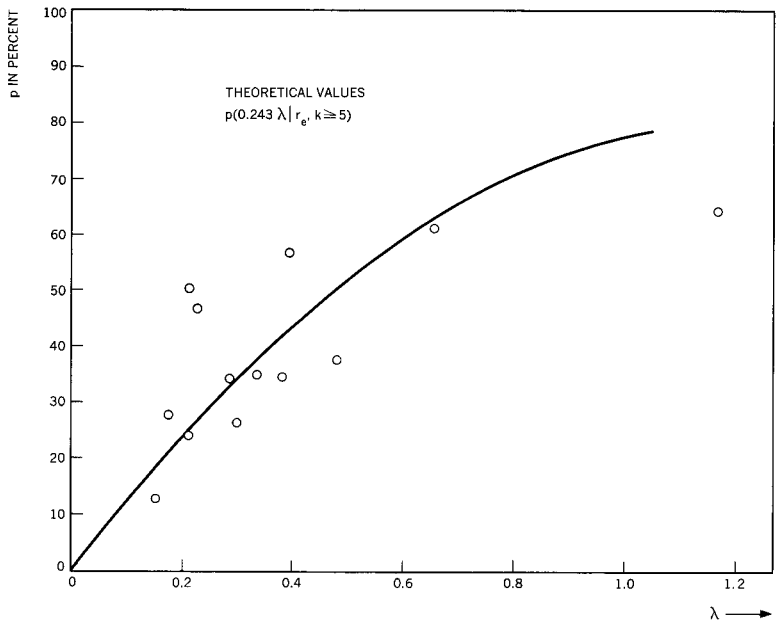
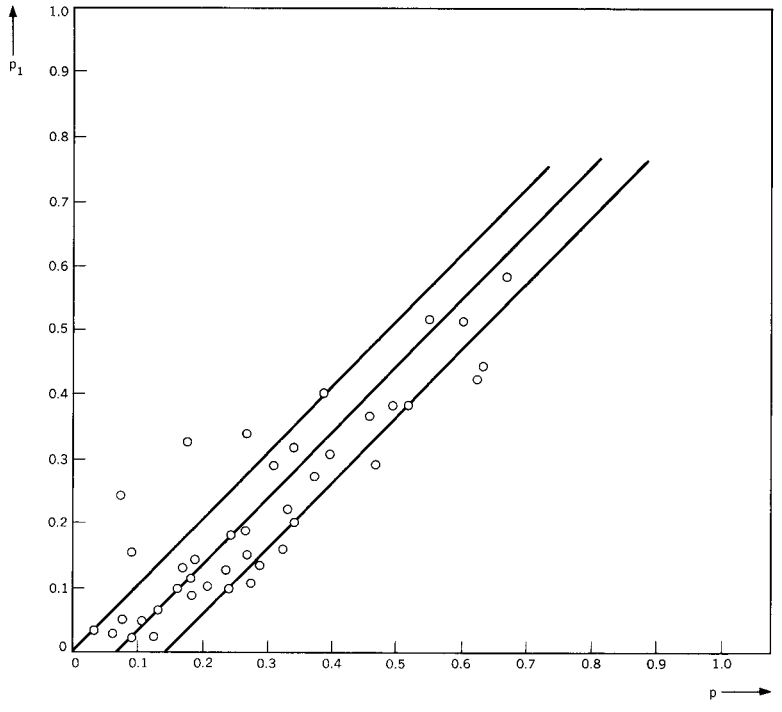


Figure 10 Total productivity (p) and productivity excluding system tasks (p_1) for homogeneous blocks less than thirty minutes



by the inaccuracies in allocating program steps to adjacent time blocks.

Recall that we treat the system tasks as an additional storage region and that (as shown in Figure 8) a single region is responsible for about 7.5 percent of productivity. If our concept is correct, then the mean horizontal distance of the empirical points in Figure 10 from the line passing through the origin with slope of one should be about 7.5 percent. Two additional parallel lines are drawn at distances of 7.5 and 15 percent. The line in the center corresponds well to a regression line, and it appears that the horizontal distance hardly ever exceeds 15 percent. Thus, the treatment of the system tasks as an additional region fits well into the total picture of the model.

expected long-term productivity

Thus we may say that the theoretical model (after correcting the computing intensity) yields an approximation of the actual processes. Before the corrected model may be used as the basis for planning, however, it may be desirable to test the model for longer time intervals, such as on a monthly basis, because planning decisions are usually based on monthly (or even annual) work loads. Accordingly, we should be interested in the mean productivity over these longer time intervals. If measurements for longer time intervals are made, we expect that the dispersion of the observations about the mean value of productivity will be considerably reduced. The property of reduced variance is highly desirable for planning purposes.

model refinement

Aside from applications to planning, further development and refinement of the model itself are of interest because the degree of correspondence between reality and model can be improved. The fact that the correlation factor for I/O times is almost four indicates that certain parts of the phenomenon are unexplained. Of course, this is not necessarily bad. In fact, the strength of cybernetic models rests in the fact that a relatively simple model structure is combined with a "variety generator." This means that the unexplained part of the phenomenon causing this high variety is summed up by certain correction factors and their probability distributions. The strength of cybernetic model building rests on this approach.[2]

Nevertheless, it seems worthwhile to ask whether the degree of explanation afforded by our planning model could not be increased with moderate effort. The fact that effective I/O times are substantially greater than actual I/O times might be explained in various ways. Recall that the theoretical model views the processes in each region of main storage as a chain of alternating I/O phases and compute phases with intervening wait times. Also a key simplifying assumption of the theoretical model is that each I/O phase that intervenes between two compute phases requires

only a single channel. As soon as the single-channel operation has been completed, the region is ready for computing. In reality, several channel operations may be required in succession or in parallel. This means that queueing for channels occurs more frequently than the theoretical model allows, even if the number of channels exceeds the number of regions. Thus the single I/O phase between two compute phases in the theoretical model is replaced by a chain of input/output phases with intervening queuing for channels.

If we insist on a single-channel model, the chain of I/O phases and queuing times must be reinterpreted as the single I/O phase of an equivalent one-channel model. It is now clear that this artificial I/O time of the single-channel model is substantially greater than the sum of the actual productive I/O times. This may well explain the large correction factor of almost four.

These considerations thus point the way to refining the model. First, it is necessary to observe the number of channels actually used during an I/O phase that intervenes between two compute phases and to study the probability distribution of the numbers of channels. Since channels are normally assigned to categories of I/O devices in a noninterchangeable manner, assignments would have to be made by category of channel.

Simulation based on a refined planning model would proceed as follows. For each I/O phase, take a sample of the channels used by category. Each selected channel is supplemented with a sample of channel time. The simulation then proceeds to implement the program and determine the intervening queuing times. Productivity curves similar to those of the theoretical model could be constructed by simulation. Correspondence with reality can then be tested. Refining the model this way may be substantially more detailed than the theoretical model, but substantially less detailed than a simulation of actual computer programs. Refinement can also improve bottle-neck problems that may exist among the channels and that cannot be handled by the theoretical model.

The existence of actual bottle-necks, in this sense, may be another explanation for long queuing times for channels and the resulting extension of I/O phases. Implementation of these ideas of more refined model construction must be the subject of further research.

some practical considerations

To maintain the validity of the basic model structure of Equation 2, it is necessary to detect significant changes of parameter values (especially γ) by continuous measurement and control. Recent developments in hardware and software monitors reduce the required experimental work load as compared to the present

study. Consider in turn the quantities that enter Equation 2. The number of channels k in the system is known, the effective number of regions $\bar{r}$ can be found by console inquires or by an accounting routine. A hardware or software monitor may be used for direct measurement of CPU busy time, CPU wait time, and channel busy times during the period of observation. Mean productivity $\bar{p}$ is estimated by the ratio of CPU busy time to the sum of busy time and wait time. Mean computing intensity $\bar{\lambda}$ may be estimated to a sufficient degree of approximation by the ratio of CPU busy time to the sum of channel busy times. (Rigorously speaking, this constitutes something of a departure from Gaver's definition of $\bar{\lambda}$, which results in a slightly different correction factor γ). At this stage, the correction factor γ may be determined by solving Equation 2.

A breakdown of total I/O time (channel time) by type of I/O device (tape, disk, drum) is most helpful when changes in peripheral equipment are contemplated. This breakdown may be obtained by measurement, such as the output of the System Management Facilities (SMF) of OS/360.

Short-range system planning

We now describe the use of the fully validated model for near-term planning of about one year. First we need the following hours-per-month work load estimates for each system alternative under consideration:

- u productive CPU time
- v sum of productive I/O times
- B'_m maximum operating time of system

From these quantities, we may deduce the following estimate of mean computing intensity:

$$\bar{\lambda} = \frac{u}{v}$$

Obviously, the given work load can be accommodated by the system if and only if the minimum CPU productivity can be expressed as

$$\bar{p}^* = \frac{u}{B_m}$$

Similarly, the required minimum I/O productivity per channel is given by

$$\bar{q}^* = \frac{v}{kB_m}$$

By contrast, the maximum obtainable productivity of the configuration in question is obtained from the corrected mean CPU productivity given by Equation 2

Table 4 System planning table

CPU	*r*	*k*	*I/O*	*Cost*	*u*	*v*	$\bar{\lambda}$	*p**	r_e	*p*	$p \geq p^*$	*Status*
C_1	5	≥ 5	A		40.8	116.0	0.35	0.27	4	0.30	Yes	Present load
C_1	5	≥ 5	A		90.0	232.0	0.39	0.60	4	0.34	No	Planned load
C_1	9	9	A		90.0	232.0	0.39	0.60	8	0.65	Yes	

Arbitrary assumption of future workload for one-shift operation and one CPU
$p = g(0.243\ \bar{\lambda}/r_e, k)$, achievable productivity from Gaver model
p^*, required minimum productivity based on maximum operating time of 150 hours per month

$$\bar{p} = g(\gamma\bar{\lambda}|\bar{r}_e, k)$$

In that equation, the symbol r_e designates the estimated mean effective number of storage regions. The system has sufficient capacity if the means of the maximum and minimum CPU productivities are related as follows:

$$\bar{p} \geq \bar{p}^*$$

There is a corresponding test for channel productivities.

short-range planning examples

To model system planning, we begin by enumerating system alternatives. The capacity test is then applied to each alternative. Systems with insufficient capacity are excluded; the remaining alternative configurations are evaluated on a cost basis. Unless criteria other than cost carry greater weight, the most cost-effective system may be selected directly, as shown by the example in Table 4. Some of the numbers are arbitrary and are intended for illustrative purposes only. The first four columns of Table 4 describe system alternatives. In addition to the type of CPU, the description includes the number of storage regions and channels, as well as one I/O configuration, A.

The first line in Table 4 reflects the assumed present condition of an existing system. The second line corresponds to a projected work load for the identical system. The result of the capacity test is negative. A proposed alternative system of line three, with its substantially increased number of regions, has sufficient capacity for the planned work load. This evaluation is repeated for all candidate systems, after which the minimum cost alternative is selected.

As shown in Table 4, each alternative has its own work load parameters u and v. It should be emphasized that these purely productive times can be estimated much more easily than system times or throughput. Quite generally, the effective number of storage regions has been assumed to be one region less than the nominal number of regions. (These and other assumptions would have to be kept up to date empirically by continuous monitoring os systems performance.)

Increasing the effective number of storage regions would seem to be a relatively simple and inexpensive way of increasing productivity. By good organization of the flow of work into the computing center and by carefully scheduling the queue of received jobs, the effective number of regions can be made to approach closely the theoretical number of regions. In our observation period, the average occupancy of storage regions was 2.6 (out of 4). Although some follow-up tests have shown that the mean occupancy had increased, it appears that significant productivity reserves remain in this area. We now turn to two further planning examples based on the conditions of the day of observation.

Long-range planning

Our model was conceived primarily for long-range system planning. The following is an outline for using a macromodel for this purpose. Future system alternatives tend to be ill-defined for long time horizons, and it is frequently argued that long-range systems planning is illusory. This argument applies to long-range planning generally. Therefore, we use the following general rule for deciding whether long-range planning is feasible. Planning, because it is anticipatory decision making, means choosing among alternative courses of action. If the alternatives cannot be foreseen with the degree of clarity required to define a choice, the time for deciding has not yet come. The decision (as well as the planning) should be postponed.

On the other hand, as soon as possible choices can be distinguished, planning is called for. This does not imply, of course, that the distinguishable system alternatives can be described with precision. The critical system parameters normally are subject to uncertainty, which does not impair the application of well-established theory on decison making under uncertainty. It is assumed, therefore, that a list of distinguishable system alternatives can be structured at the appropriate time, and that system parameters can be forecast with appropriate error bands. According to Table 4, we are particularly concerned with the work load parameters u and v (and thereby λ) for an arbitrary reference system for all system alternatives. Thus it is expedient to make the existing configuration the reference system and to begin with a projection of the mix constant λ. Instrumental monitoring of λ (as described later), trend extrapolation, and subjective tempering by foreseeable changes in computing intensity should yield a workable forecast and error band.

A forecast of an absolute measure of work load (say v as related to data volume) is far more difficult. In fact, we propose to modify somewhat the planning approach of Table 4 in view of the forecasting problem.

Table 5 Maximal system throughput

System alternative	CPU	r	r_e	k	I/O	Hardware cost	u_0/u	λ	p	T/T_0
0	System 360–65	6	5	5	A		1	λ_0	p_0	1

Rather than minimize system cost for a given work load, as in Table 4, we now choose to maximize system throughput for a given ceiling of hardware cost. System throughput is defined as follows. Let W designate the system elapsed time required to process the work load with parameters u and v. Throughput T is defined relative to the reference system (subscript zero) by

$$\frac{T}{T_0} = \frac{W_0}{W}$$

the Gaver productivity p is related to W by

$$p = \frac{u}{W}$$

so that

$$\frac{T}{T_0} = \frac{u_0}{u} \cdot \frac{p}{p_0} \tag{3}$$

Note that computation of the throughput ratio requires only the knowledge of the mix constant λ, the CPU factor u_0/u, and the configuration parameters. No absolute measures of work load enter Equation 3. The new planning sheet for determination of the throughput–maximal system (for a given ceiling of hardware cost) is shown as Table 5. Error bands of parameters are treated in accordance with the methodology of decision making under uncertainty.[3]

For purposes of illustration, a throughput comparison of four systems is given in Table 6. For simplicity, it has been assumed that all systems have an identical I/O configuration and job mix so that $\bar{\lambda}$ varies inversely with the CPU factor.

Our methodology requires frequent evaluation of the Gaver function as given in Equation 2, wherein complex recursive techniques are used to arrive at numerical values of the mean productivity $\bar{p}$. G. Diruf[4] showed that the nonexplicit Gaver function of Equation 1 is identical to the following analytical relationship, provided that CPU and I/O times per segment are exponentially distributed:

$$\bar{p}(k, r, \bar{\lambda}) = \frac{b_1(k\bar{\lambda}) + b_2(k\bar{\lambda})^2 + \cdots + b_r(k\bar{\lambda})r}{1 + b_1(k\bar{\lambda}) + b_2(k\bar{\lambda}) + \cdots + b_r(k\bar{\lambda})^r}$$

Table 6 Throughput comparison for a given I/O configuration and job mix

Quantity	System				
	System/360 *Model* 65	A1		A2	A3
CPU factor (u_0/u)	1.00	4.00	–	6.00	3.00
k	6.00	11.00	–	11.00	11.00
r	6.00	10.00	12.00	12.00	12.00
$\bar{\lambda}$	0.79	0.20	0.20	0.13	0.25
$\tilde{p}$	0.79	0.45	0.51	0.35	0.65
T/T_0	1.00	2.28	2.58	2.62	2.47

where $b_n = \prod_{i=1}^{n} a_i$

and $a_1 = \begin{cases} 1 & \text{for } 1 \leq i \leq r-k \\ \dfrac{r-i+1}{k} & \text{for } r-k < i \leq r \end{cases}$

Future research

To cover the full range of system alternatives, our planning system should be generalized to include models of multiprocessors, multishift operation, and other system configurations. A more detailed investigation of peripheral equipment seems desirable. In addition, the overhead phenomenon should be treated explicitly. Strictly speaking, any throughput comparison between systems must be based on *net productivity* $\tilde{p}$ excluding the fraction α of capacity consumed by the supervisor, which presumably assumes special significance for multiprocessors. If a system employs n (symmetrical) processors with r regions and k channels per processor, the model of net productivity appears in the following form:

$$\tilde{p} = (1 - \alpha) \cdot p(\gamma\bar{\lambda}|n, r, k)$$

Here p designates gross productivity in the sense used so far. Simulation models of the buffering effect of multiprocessors have already been completed, and tables of gross productivity p have been drawn up. The investigation of supervisory loss α, especially its dependence on the configuration, awaits further experimentation and research. Certain other aspects of multiprocessing (e.g., reliability and maintainability) remain outside these models.

Summary comment

Existing empirically founded techniques for timing individual jobs are not suitable for long-range planning. The main reasons

are: they are tied to an existing system structure; necessary detail on individual jobs is not available on a long-range basis; and the types of jobs are subject to considerable uncertainty. To improve this situation, we suggest a planning technique for many system alternatives. In addition, this technique should require only relatively general information about the work load at the planning horizon. The basis for the model is a probability-based macromodel of multiprogramming by Gaver.

Since system productivity (CPU utilization) is of central interest, our planning model has been conceived to forecast productivity and throughput for numerous system alternatives. Certain corrections and further developments of the theoretical model are required to validate the model for planning applications. An initial test of validity and modifications of the planning model described are based on experiments using System/360 Model 65.

We have outlined approaches to short-range and long-range forecasting of demand for data processing services. A productivity model is supplemented by simulation models of the buffering effect of multiprocessing. Modeling of supervisory overhead awaits further research.

CITED REFERENCES AND FOOTNOTE

1. D. P. Gaver, "Probability models for multiprogramming computer systems," *Journal of the ACM* **14,** No. 3, 423–438 (1967).
2. S. Beer, *Decision and Control*, John Wiley and Sons, New York, New York (1966).
3. F. Hanssmann, *Operations Research Techniques for Capital Investment*, John Wiley and Sons, New York, New York (1968).
4. G. Diruf, *A Two-Stage Macro-Model for Multiprocessing.* (This unpublished Ph.D. thesis bases on work done at the University of Munich may be obtained as an IBM report from IBM Germany, Boeblingen, Germany.)