

Technical Report CS 75016-R  
A PROCESSOR UTILIZATION MODEL  
FOR A  
MULTIPROCESSOR COMPUTER SYSTEM

Richard E. Nance  
Department of Computer Science  
Virginia Polytechnic Institute and State University

and

U. Narayan Bhat\*  
Department of Industrial Engineering  
and Operations Research  
Southern Methodist University

August 1975

\*Research work by this author was supported by DCED/ONR contract No. 0014-75-C-0517, NRO42-324. Reproduction in whole or in part is permitted for any purpose of the United States Government.

## ABSTRACT

A processor utilization model for a simplified multiprocessor computer system is developed. Jobs are assumed to arrive according to a general input process, and each job is assigned randomly to an available processor. A finite capacity input buffer is used if no processor is available. The mathematical model is based on the busy period analysis, and two utilization measures are derived:

- (1) processor utilization when the system is busy (the fraction of processor occupation time during a busy period), and
- (2) global processor utilization (the fraction of processor occupation time during a busy cycle).

Additionally, the arbitrary time state probability distribution is obtained and serves as the basis for the above measures in addition to others. Several approximations enable the development of a computational model from the mathematical model. Experimentation with the computational model reveals the sensitivity of the model to variability in the arrival process. Comparison of 2-processor and 4-processor systems from the operator perspective indicates a qualified preference for the behavior of the 2-processor system. This preference must be carefully interpreted since processor costs, the increase in overhead with an increase in processors, and behavioral variables reflecting the user perspective are excluded.

Keywords: multiprocessor, processor utilization, finite buffer capacity, computational model, busy cycle, experimental comparison.

Computing Reviews category: 4.32

## INTRODUCTION

Interest in multiprocessing computer systems, i.e. systems utilizing two or more interconnected processing units in order to execute two or more different programs or tasks simultaneously [1, p. 305], began quite early. A 1963 paper by Critchlow [2] contains some 44 references. Both General Electric (GE 645) and Control Data (CDC 6500) introduced multiprocessor systems in the mid 1960's; however, interest in these systems has increased markedly in recent years. This interest has been stimulated by several interrelated developments:

- (1) the reduction in the cost of main frames,
- (2) the rapid emergence of the mini- and micro-processors accompanied by the perspective of distributed computer systems, and
- (3) the increased motivation for sharing of resources through computer networks using high speed data communications.

The recent compilation of material edited by Enslow [16] gives a more precise definition of a multiprocessor system and explains the variations in design and topologies among such systems. Regardless of the topology of a multiprocessor system, e.g. a large duplex main frame located in the same room, a pipeline or array processor utilizing extensive concurrent operations or a set of mini-computers distributed geographically and organizationally throughout a major firm, efficiency is still the major issue.

The research described herein explores the processor utilization in a multiprocessing system. We are careful to emphasize that, although efficiency and utilization are closely related, a distinct difference between the two must be recognized. Efficiency is a measure based on the use of a computer system resource, e.g. a central processing unit (CPU), in the processing of particular user tasks. Utilization is a measure of the period of use of a resource for either user or system tasks. Consequently, a resource could be heavily utilized but very

inefficiently used, e.g. a CPU in which the operating system tasks demand 80 percent of the processing time. We believe that a measure of efficient use of a system resource must include utilization as one component, i.e. high efficiency necessitates high utilization but not the converse.

Within the limitations of processor utilization as a behavioral measure, we investigate the reaction to different job arrival distributions and compare the behavior of 2-processor and 4-processor systems. A mathematical model of a simplified multiprocessor system is constructed and, by employing several approximations, we develop a computational model to enable experimental work.

#### Previous Modeling Treatments

Prior models of performance in multiprocessing systems are generally directed toward solving one of two problems: (1) the testing of various scheduling disciplines, possibly with the intent of identifying the most preferred among a set of alternatives [4,5,6] or (2) the description of the memory contention (interference) arising from the use of a common finite memory concurrently accessible by the group of processors [7,8,12,15].<sup>1</sup> Notable exceptions are the paper by Coffman [10], which develops relationships between the number of jobs in the system, the number of processors and the loading (the number of tasks per job), and the recent work by Kafura and Shen [3], which gives a combined treatment of independent storage capacities for each processor and the scheduling disciplines. Also, more recent performance models have addressed particular architectural configurations, e.g. see Ramamoorthy and Kim [11] and Ramamoorthy and Li [18].

Among the cited works above are examples of each of the techniques applied to performance modeling:

- (1) deterministic or graph models [3,6],

- (2) probabilistic or queue-theoretic models [10], and
- (3) simulation studies [8,12].

The second technique is used in this work, and a computational model is derived for experimental comparisons.

### Objectives and Model Characteristics

Our objective is to focus on the processor utilization during the long term operation of the system. Utilization provides a measure of the relative use of the available processors during periods of user demand. This measure reflects the operator-oriented perspective, which we distinguish from the user-oriented perspective in an earlier paper [13, pp. 221-222]. In accomplishing our objective, we develop a mathematical model of processor occupation time based on an embedded Markov chain analysis coupled with a state visitation process during transitions among states comprising the embedded chain. A computational model is derived to test the model sensitivity to the job arrival distribution and the input buffer capacity.

Figure 1 provides a sketch of the simplified model of a multiprocessing system. Arriving jobs are assigned by a "dispatcher" immediately to a "free" processor. If all processors are busy, jobs are assigned to an input buffer with finite storage capacity. Arriving jobs finding the input buffer filled are "lost", e.g. when the input buffer is saturated, all input terminals are temporarily blocked from further transmission. Jobs are assigned to available processors according to an appropriate scheduling rule, and the job is processed to completion without interruption. Internal memory capacity is assumed sufficient for all job/processor combinations. Completed jobs are released from the system, i.e. once a job is assigned to a processor all remaining input/output functions are performed by that processor (either individually or allocated to

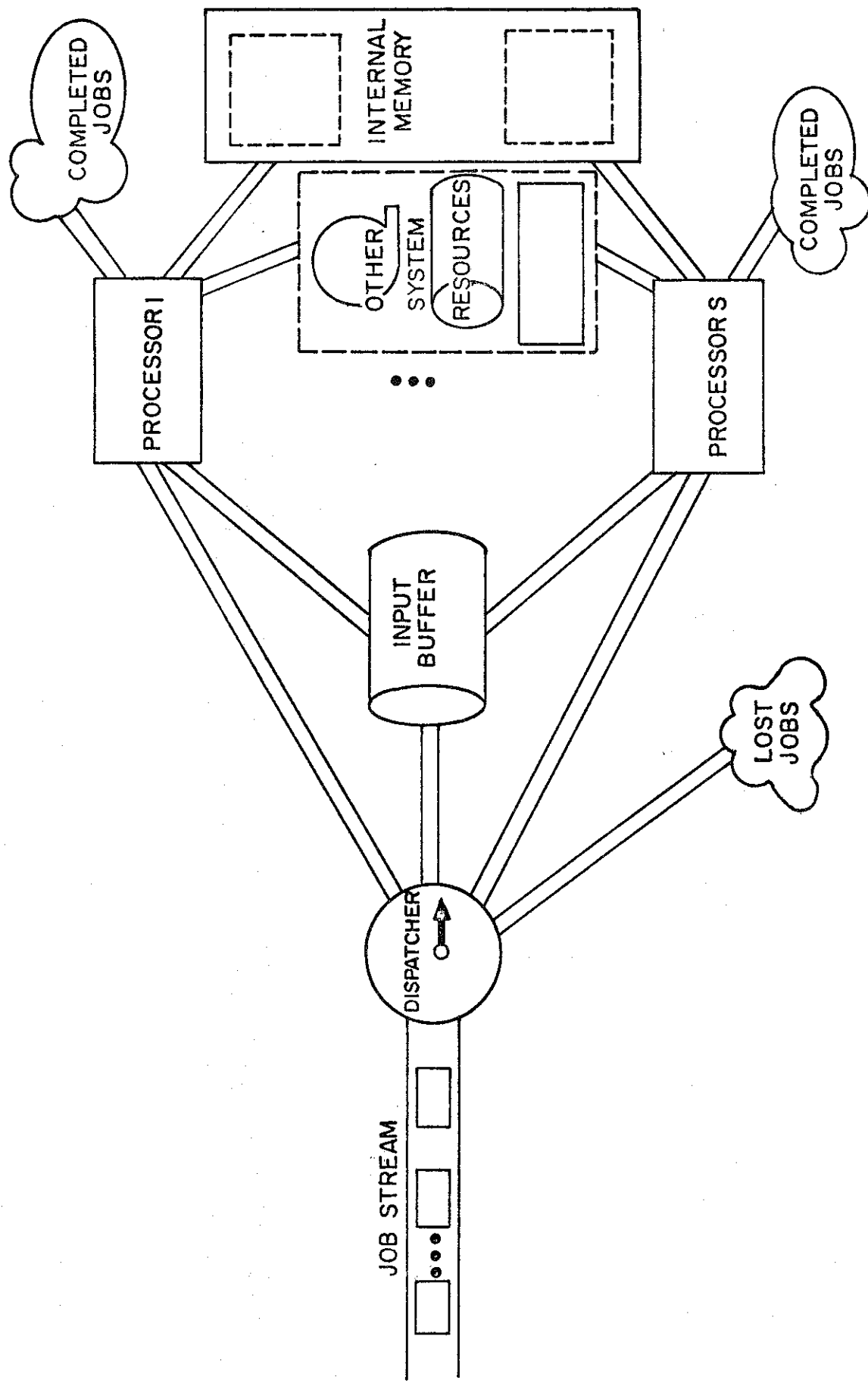


Figure 1. Sketch of Simplified Multiprocessing System

an I/O processing unit). No hardware or software failures are considered since we are investigating the relationships among the job arrival process, the input/buffer capacity and the number of processors. Channel capacity and interference problems are also ignored.

## MATHEMATICAL MODEL

### Definitions and Assumptions

A busy period (BP) is defined as the time interval during which the system is continuously busy. A busy period followed by an idle period (during which no jobs are processed) is defined as a busy cycle (BC). Viewing the system on a time axis, it appears to go through a sequence of busy cycles. Under stationary conditions and in steady-state, the mean value characteristics derived for the busy cycle represent the corresponding mean value characteristics for the system behavior.

Depending of the number of jobs in the system (assigned to processors and in the input buffer) during a busy period, one or more processors are utilized. The following processor utilization measures are defined:

$$\begin{aligned} & \text{processor utilization with the system busy (PU}_B\text{)} \\ & = \frac{\text{processor occupation time}}{\text{system occupation time during a busy period}} \end{aligned} \quad (1)$$

$$\begin{aligned} & \text{processor utilization (PU)} \\ & = \frac{\text{processor occupation time}}{\text{mean length of busy cycle}} \end{aligned} \quad (2)$$

Processor occupation time is defined as the period during which the processor is occupied by a job during a busy cycle.

With no restrictions on the utilization of individual processors and the assumption of identical processing rate, the service load is equally distributed among the processors. Let  $\rho$  be the offered service load per processor defined as

$$\rho = \frac{\text{arrival rate}}{(\text{number of processors}) \times (\text{processing rate})}$$

and  $\rho^*$  be the effective service load per processor resulting from the finite buffer capacity. Let  $P_B$  be the probability that a job encounters a filled buffer



on its arrival. Then we have

$$\begin{aligned}\rho^* &= \frac{(\text{arrival rate}) \times (1-PB)}{(\text{number of processors}) \times (\text{processing rate})} \\ &= (1-PB)\rho.\end{aligned}\quad (3)$$

The processor utilization PU can be given by  $\rho$  or  $\rho^*$  accordingly. Also, let  $p_0$  be the probability that the system is idle in the long run. An expression for  $p_0$  can be given as

$$\begin{aligned}p_0 &= \frac{\text{system idle time during a busy cycle}}{\text{mean length of busy cycle}} \\ &= 1 - \frac{\text{system occupation time}}{\text{mean length of busy cycle}}\end{aligned}$$

The two utilization measures PU and  $PU_B$  have the relation

$$\frac{PU_B}{PU} = \frac{1}{1-p_0}$$

Thus the determination of the two utilization measures requires either the information concerning the occupation time during a busy cycle or the probability of blocking PB and the probability of system idleness  $p_0$ . The value PB is obtained as the probability of a filled buffer in an arrival epoch steady state distribution, and the probability  $p_0$  is obtained as the probability of emptiness in an arbitrary time steady state distribution. Except in some special cases, the relationship between the two probability distributions is not exactly known (see, Takacs [14], Chapter 1, for the known relation for an infinite buffer capacity); therefore, the information provided by the arrival epoch distribution is not complete. Consequently, we develop an alternate procedure which determines all system utilization measures of practical significance for the operator-oriented perspective. As illustrated in a subsequent section, the method is also convenient for computational use. Additionally, the potential for application of this method to different operating system policies seems excellent.

The following assumptions are stated with regard to the basic characteristics of the system. The repetition of certain points from the previous section is simply to place all assumptions in one section.

- (1) There are  $s$  identical parallel processors in the system. Processing times for individual jobs follow an exponential distribution with mean  $1/\mu$  for each processor.
- (2) The sequence of time epochs  $t_0, t_1, t_2, \dots$  mark the arrivals of jobs. Let  $Z_n = t_n - t_{n-1}$  ( $n=1,2,\dots$ ). We assume that the sequence of random variables  $\{Z_n\}$  are distributed as

$$P[Z_n \leq t] = A(t) \quad (t \geq 0)$$

and

$$E[Z_n] = a$$

The random variables  $Z_n, n = 1,2,\dots$  are assumed to be independent and identically distributed throughout our discussion. This is done only for convenience and the extension to a state-dependent random variable presents no difficulty.

- (3) The system has an input buffer with capacity for  $N-s$  waiting jobs.
- (4) Let  $J_n$  be the number of jobs in the system just before an arrival at  $t_n$  ( $n = 1,2,\dots$ ). If  $J_n < s$ , the arriving job is randomly assigned by the dispatcher to an available processor. If  $N > J_n \geq s$ , the arriving job is assigned to the input buffer to await processing. If  $J_n = N$ , the job arrival stream is disabled (or the job is considered lost).
- (5) Once assigned to a processor, the job is completely serviced and exits the system. Any further input/output requirements of the job are accomplished by the assigned processor (either by that processor or an assigned I/O processor).
- (6) Internal memory capacity, whether shared or dedicated, is sufficient for any combination of processing tasks, and the requirements on any other

resources are reflected in the processing time of each job.

### The Processes $J_n$ and $J(t)$

Based on the above assumptions we note that the process  $\{J_n\}$  is a finite Markov chain with transition probabilities  $\alpha_{ij}$  ( $i, j = 0, 1, 2, \dots, N$ ) such that

$$\alpha_{ij} = \int_0^{\infty} dP_{ij}(x)$$

where

$$dP_{Nj}(x) = \begin{cases} e^{-s\mu x} \frac{(s\mu x)^{N-j}}{(N-j)!} dA(x) & (s \leq j \leq N) \\ dA(x) \int_0^x e^{-s\mu y} \frac{(s\mu y)^{N-s-1}}{(N-s-1)!} s\mu \binom{s}{j} [1 - e^{-\mu(x-y)}]^{s-j} e^{-j\mu(x-y)} dy & (j < s) \end{cases}$$

$$dP_{ij}(x) = \begin{cases} e^{-s\mu x} \frac{(s\mu x)^{i-j+1}}{(i-j+1)!} dA(x) & (s \leq j \leq i+1) \\ dA(x) \int_0^x e^{-s\mu y} \frac{(s\mu y)^{i-s}}{(i-s)!} s\mu \binom{s}{j} [1 - e^{-\mu(x-y)}]^{s-j} e^{-j\mu(x-y)} dy & (s \leq i, j < s) \\ \binom{i+1}{j} [1 - e^{-\mu x}]^{i-j+1} e^{-j\mu x} dA(x) & (i < s, j \leq i+1) \end{cases} \quad (5)$$

While using (4) and (5) to obtain  $\alpha_{ij}$  in a convenient form, we introduce the notation

$$\gamma_j(\delta) = \int_0^{\infty} e^{-\delta x} \frac{(\delta x)^j}{j!} dA(x) \quad (6)$$

We obtain the following expressions

$$\alpha_{Nj} = \begin{cases} \gamma_{N-j}(s\mu) & (s \leq j \leq N) \\ \binom{s}{s-j} \sum_{\ell=N-s}^{\infty} \left[ \sum_{r=0}^{s-j} (-1)^r \binom{s-j}{r} \left(\frac{s-j-r}{s}\right)^{\ell-N+s} \right] \gamma_{\ell}(s\mu) & (j < s) \end{cases} \quad (7)$$

$$\alpha_{ij} = \begin{cases} \gamma_{i-j+1}(s\mu) & (s \leq j \leq i+1; i \geq s-1) \\ \binom{s}{s-j} \sum_{\ell=i-s+1}^{\infty} \left[ \sum_{r=0}^{s-j} (-1)^r \binom{s-j}{r} \left(\frac{s-j-r}{s}\right)^{\ell-i+s-1} \right] \gamma_{\ell}(s\mu) & (s \leq i, j < s) \\ \binom{i+1}{j} \sum_{r=0}^{i-j+1} (-1)^r \binom{i-j+1}{r} \gamma_0[(r+j)\mu] & (i < s, j \leq i+1) \end{cases} \quad (8)$$

During a busy period transitions of the Markov chain  $\{J_n\}$  occur only among  $\{1, 2, \dots, N\}$ . Since  $\{J_n\}_{n=0}^{\infty}$  represents the state of the system only at arrival epochs, we must determine the number of visits to different states between arrivals in order to derive the processor occupation time. Therefore, let  $J(t)$  be the number of jobs in the system at time  $t$ . We can obtain the processor and system occupation times during a busy period in two stages:

- (1) Determine the expected number of visits of  $\{J_n\}$  to states  $1, 2, \dots, N$  during a busy period.
- (2) Determine the occupation time of the process  $J(t)$  in states  $1, 2, \dots, N$  for every visit of  $\{J_n\}$  to a particular state.

Of these, the first stage follows directly from the theory of finite Markov chains (e.g., see Kemeny and Snell [17]).

Partition the transition probability matrix  $P$  of the Markov chain  $\{J_n\}$  as follows.

$$P = \begin{pmatrix} \alpha_{00} & \alpha_{01} & \dots & \alpha_{0N} \\ \alpha_{10} & & & \\ \alpha_{20} & & H & \\ \vdots & & & \\ \alpha_{N0} & & & \end{pmatrix} \quad (9)$$

Let

$$(I-H)^{-1} = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1N} \\ v_{21} & v_{22} & \dots & v_{2N} \\ \vdots & \vdots & & \vdots \\ v_{N1} & v_{N2} & \dots & v_{NN} \end{pmatrix} \quad (10)$$

From the theory of finite Markov chains we know that the expected number of visits of the process to state  $j$  during a busy period, having initiated from state  $i$ , is given by  $v_{ij}$ .

For the second stage, we divide the transitions occurring between two consecutive arrival epochs into two cases: (i)  $J_n = j$  and  $J_{n+1} = k$  ( $> 0$ ) and (ii)  $J_n = j$  and  $J_{n+1} = 0$ .

Case i: During the inter-arrival interval,  $J(t)$  passes through the states  $j+1, j, \dots, k$ . The unconditional occupation time of  $J(t)$  in state  $r$  ( $r=j+1, j, \dots, k$ ) is  $1/\mu$ . However, these transitions are observed during an interval with mean length  $a$ ; consequently, the conditional occupation time in state  $r$  is obtained as

$$\hat{a}_{kj}(r) = \frac{a/\min(r,s)}{\sum_{m=k}^{\min(j+1,N)} [1/\min(m,s)]} = \frac{\hat{a}(r)}{d_{kj}} \quad (11)$$

$k = 1, 2, \dots, r; j = r-1, r, \dots, N,$

where  $\hat{a}(r)$  and  $d_{kj}$  are used to represent expressions in the numerator and denominator of (11) respectively. The above expression results from the independence of the transition  $j \rightarrow k$  of the process  $\{J_n\}$  and the inter-arrival period  $Z_{n+1}$  and the fact

that within this interval  $J(t)$  is a pure death process. In such a process with death rate  $\mu$  per job, the mean occupation time in state  $r$  is  $1/r\mu$ . Thus, when the process  $J(t)$  goes through a transition  $(j+1) \rightarrow k$ , with probability  $\alpha_{jk}$ , the fraction of time state  $r$  is occupied is obtained as

$$[1/\min(r,s)]/d_{kj}$$

Hence the expression (11) above is derived.

Case (ii): At the conclusion of a busy period, i.e., when  $J_n = j$ ,  $J_{n+1} = 0$ , the amount of time required for first passage to zero is dependent on the initial state  $j$ ; consequently, the arguments used in Case (i) to obtain the state occupation times do not hold. Retreating to basic arguments, we denote by  $Y_r$  the occupation time in state  $r$ . Note that the distribution of  $Y_r$  is a conditional exponential with parameter  $r\mu$  such that neither  $Y_r$  nor  $\sum_{i=1}^{j+1} Y_i$  exceed the length of the inter-arrival period  $Z_n$ . Let  $c_j(y)$  be the p.d.f. of  $\sum_{i=1}^{j+1} Y_i$ . We have for  $Z_n = z$  and  $0 < y < z$

$$c_j(y)dy = \begin{cases} (j+1)[1-e^{-\mu y}]^j e^{-\mu y} \mu dy & 0 \leq j < s \\ \int_{x=0}^y e^{-s\mu x} \frac{(s\mu)^{j-s+1} x^{j-s}}{(j-s)!} s[1-e^{-\mu(y-x)}]^{s-1} e^{-\mu(y-x)} \mu dx dy & s \leq j < N \\ \int_{x=0}^y e^{-s\mu x} \frac{(s\mu)^{j-s} x^{j-s-1}}{(j-s-1)!} s[1-e^{-\mu(y-x)}]^{s-1} e^{-\mu(y-x)} \mu dx dy & j=N \end{cases}$$

Therefore we obtain

$$\bar{a}_j(r) = \int_0^\infty \left[ \frac{\int_0^z ye^{-r\mu y} r\mu dy}{\int_0^z c_j(y) dy} \right] dA(z) \quad (12)$$

$j = r-1, r, \dots, N.$

Clearly, the evaluation of  $\bar{a}_j(r)$  presents some difficulty. For the purposes of numerical investigations we suggest the following approximation  $\tilde{a}_{0j}(r)$  using unconditional means. Let

$$a'_j = \min \left\{ \frac{1}{\mu} \sum_{i=1}^{\min(j+1, N)} [1/\min(i, s)], a \right\} \quad (13)$$

$$j = 0, 1, \dots, N.$$

and write

$$\tilde{a}_{0j}(r) = \frac{a'_j / \min(r, s)}{\sum_{m=1}^{(N, j+1)} [1/\min(m, s)]} = \frac{\tilde{a}_j(r)}{d_{1j}} \quad (14)$$

where the numerator of (14) is denoted as  $\tilde{a}_j(r)$ . Clearly  $\tilde{a}_{0j}(r)$  overestimates  $\bar{a}_j(r)$  in case (ii). However, for moderate to large values of  $j$ ,  $\alpha_{j0}$  is expected to be very close to zero, and the effect of approximation is presumed negligible.

#### Processor Utilization

From the results derived in the last section, we develop expressions for the expected state occupation times  $E[S_r]$  during an expected busy cycle  $E(BC)$ . By definition

$$E(BC) = \sum_{r=0}^N E(S_r) \quad ; \quad (15)$$

also, the mean busy cycle can be derived using the mean first passage times  $(v_{ij})$  of (10). Noting that these first passages are conditional on the busy cycle extending beyond the first inter-arrival period, we write

$$E(BC) = a [1 + \gamma_0(\mu) \sum_{j=1}^N v_{1j}] \quad (16)$$

where  $\gamma_0(\mu) = \int_0^\infty e^{-\mu x} dA(x)$  is the probability that the busy cycle extends beyond the first inter-arrival period.

Processor utilization requires the determination of individual state occupation times. Let  $E_c(S_r)$  be the occupation time of state  $r$  conditional on the busy cycle extending beyond one transition interval. Considering the number of visits of the process  $\{J_n\}$  to different states and the state occupation times of the process  $\{J(t)\}$  between transition epochs, we obtain (using the approximation suggested earlier)

$$\begin{aligned}
 E_c(S_r) &= \sum_{j=\max(r-1,1)}^N v_{1j} \left[ \sum_{k=1}^r \alpha_{jk} \hat{a}_{kj}(r) + \alpha_{j0} \tilde{a}_{0j}(r) \right] \\
 &= \sum_{j=\max(r-1,1)}^N v_{1j} \left[ \hat{a}(r) \sum_{k=1}^r \left[ \alpha_{jk}/d_{kj} \right] + \tilde{a}_j(r) \alpha_{j0}/d_{1j} \right] \\
 & \qquad \qquad \qquad r = 1, 2, 3, \dots, N. \quad (17)
 \end{aligned}$$

Removal of the condition on state occupation times results in

$$E(S_r) = \gamma_0(\mu) E_c(S_r) \qquad r = 2, 3, \dots, N. \quad (18)$$

The expression for  $E(S_1)$  must also include the possibility of termination of the busy cycle with only one service. Thus we get

$$E(S_1) = \bar{a}_0(1)[1-\gamma_0(\mu)] + [a + E_c(S_1)]\gamma_0(\mu) \quad (19)$$

where  $\bar{a}_0(1)$  is to be obtained from (12). As an approximation for  $\bar{a}_0(1)$ , we may use  $a'_0$  given by (13).

Expressions for the expected processor occupation time ( $E_{pot}$ ) and the expected system occupation time ( $E_{sot}$ ) follow directly

$$E_{pot} = \sum_{r=1}^{s-1} \frac{r}{s} E(S_r) + \sum_{r=s}^N E(S_r) \quad (20)$$

$$E_{sot} = \sum_{r=1}^N E(S_r) \quad (21)$$

The two measures of processor utilization suggested earlier can be given as



(1) processor utilization with the system busy

$$PU_B = \frac{E_{pot}}{E_{sot}}$$

(2) processor utilization

$$PU = \frac{E_{pot}}{E(BC)}$$

In the second expression  $E(BC)$  is obtained as in (16). Because of an approximation in (13) and (14), when the mean inter-arrival time is smaller than the mean processing time (which is possible with more than one processor),  $E(BC) \approx E_{sot}$ . Although more exact evaluations of (12) are possible in order to maintain the distinction between  $E(BC)$  and  $E_{sot}$ , we feel that the additional information that can be derived is not justifiable (especially considering the likelihood of introducing error in computing the ratios of the integrals).

As a result of the approximation in (12), the numerical values obtained here slightly overestimate the utilization measures. When the arrival rate of jobs relative to their processing rate is low, the blocking probability is negligible and the processor utilization  $PU$  is very close to  $\rho$  (defined earlier). Therefore, as a correction for our utilization measure, we write

$$PU = \min[E_{pot}/E(BC), \rho] \quad (22)$$

The steady state distribution of the number of jobs in the system at an arbitrary time point follows easily from our results. We have

$$p_r = \frac{E(S_r)}{E(BC)}, \quad r = 1, 2, 3, \dots, N$$

$$p_0 = 1 - \sum_{r=1}^N p_r. \quad (23)$$

Furthermore, using the discussion following equation (3) we determine the probability of blocking as

$$PB = 1 - PU/\rho \quad (24)$$

## COMPUTATIONAL MODEL

One consequence of the translation of the mathematical model into a computational model has been noted, i.e. the approximation of

$$\bar{a}_j(r) = \int_0^\infty \left( \frac{\int_0^z y e^{-r\mu y} r\mu dy}{\int_0^z c_j(y) dy} \right) dA(z)$$

$$j = r-1, r, \dots, N$$

by  $\tilde{a}_{0j}(r)$  where

$$\tilde{a}_{0j}(r) = \frac{a'_j / \min(r, s)}{\min(N, j+1) \sum_{m=1} [1/\min(m, s)]}$$

and

$$a'_j = \min \left\{ \frac{1}{\mu} \sum_{i=1}^{\min(N, j+1)} [1/\min(i, s)] \right\}, \quad a \}$$

Three other aspects of the computational model deserve mention. The first involves the computation of the values  $\gamma(\delta)$  given in (6) as

$$\gamma_j(\delta) = \int_0^\infty e^{-\delta x} \frac{(\delta x)^j}{j!} dA(x)$$

For any arrival process we compute all values  $j = 1, 2, \dots, n$  such that

$$\gamma_j(\delta) > \epsilon_1$$

with  $\epsilon_1$  a prescribed error bound.

The second aspect is concerned with the calculation of the infinite sums contained in (7) and (8), e.g. the second expression in (7).

$$\alpha_{Nj} = \binom{s}{s-j} \sum_{\ell=N-s}^{\eta} \left( \sum_{r=0}^{s-j} (-1)^r \binom{s-j}{r} \left(\frac{s-j-r}{s}\right)^{\ell-N+s} \right) \gamma_{\ell}(s\mu)$$

$j < s$

where  $\eta$  is such that

$$\left\lfloor \frac{s}{2j} \right\rfloor \gamma_{\eta+1}(s\mu) < \varepsilon_2,$$

where  $\lfloor x \rfloor$  is the greatest integer  $\leq x$  (the floor function).

This rather simple single term cutoff is justified by the fact that the  $\gamma_j(s\mu)$  values are probabilities and are strictly monotone non-increasing with increasing values of  $j$ . The truncation term  $\eta$  is computed only once since we can easily show that the contribution of the similar subexpression in (8) cannot exceed  $\varepsilon_2$ .

The final aspect of the computational model concerns the determination of the matrix  $(I-H)^{-1}$ . Observing that the probability transition matrix has the lower Hessenberg structure, i.e.

$$\begin{pmatrix} \alpha_{00} & \alpha_{01} & & 0 \\ \alpha_{10} & \alpha_{11} & \alpha_{12} & \\ \vdots & \vdots & & \\ \alpha_{N-1,0} & \alpha_{N-1,1} & \dots & \alpha_{N-1,N} \\ \alpha_{N0} & \alpha_{N1} & \dots & \alpha_{NN} \end{pmatrix}$$

We use a Gaussian elimination method with row pivoting for solving the linear system in a very efficient manner.

Empirical results are obtained from FORTRAN programs developed and executed using the FTN compiler on a CDC 6700 and the G-Level FORTRAN IV compiler on a dual IBM S370/158 system. All programming was done by the authors except for the Gaussian elimination routine provided by Professor James E. Kalan.

## EXPERIMENTAL RESULTS

Experiments with the model focus on four behavioral variables:

- (1) the arbitrary time state probability distribution, which cannot be obtained by other approaches,
- (2) the expected busy cycle, and
- (3) the processor utilization measures  $PU$  and  $PU_B$ .

Both (2) and (3) can be obtained easily from (1); yet each offers an added insight into the total behavior. We also provide the expected number of jobs in the input buffer.

Our intent is to determine the behavior of the multiprocessor model under three conditions:

- (1) differing variability levels in the job inter-arrival time distribution -- using an Erlangian  $(k, \lambda)$  with  $\lambda = .5$  and  $k = 2, 4$  and  $8$  with coefficient of variation (C. V. =  $100\sigma/\mu$ ) values of  $70.7$ ,  $50.0$  and  $35.3$  respectively;
- (2) increasing demand on a system with a fixed number of homogeneous processors, each having an identical processing rate; and
- (3) testing the relationship between the number of processors and individual processor capability in a homogeneous multiprocessor system.

Figures and tables are used to summarize the results.

Table 1 provides indications of the effect of a highly variable inter-arrival distribution on a system with two processors ( $s$ ) and a buffer capacity  $(N-s)$  of six jobs. In the three cases shown, the expected inter-arrival time is doubled, with the result that the offered load per processor is halved. The effect on the probability of blocking ( $PB$ ) and the expected busy cycle  $E(BC)$  are quite dramatic. The third case shows a decrease in  $E(BC)$  of more than two orders of magnitude indicating that idle periods are occurring far more frequently than in the second case. The average number of jobs in the input buffer and the processor utilization values ( $PU_B$  and  $PU$ ) show that within a busy period activity remains high

Erlangian Distribution of Interarrival Times	Offered Load per Processor	Probability of Blocking	E(BC) Expected Busy Cycle	Avg.No.Jobs in Input Buffer	PU <sub>B</sub> Processor Utilization, System Busy	PU Processor Utilization
Case 1						
k = 2 C.V. = 70.7	5.00	.80	958623	5.86	1.00	1.00
Case 2						
k = 4 C.V. = 50	2.50	.60	442075	5.63	1.00	1.00
Case 3						
k = 8 C.V. = 35.3	1.25	.21	2098	4.23	.99	.98

$\lambda = .5$   
 $\mu = .025$

$N-s = 6$   
 $s = 2$

Table 1. Behavior with Differing Variability Levels in the Job Inter-arrival Time Distribution

in all three cases. However, a high variability in inter-arrival times (the third case) precipitates far more frequent, although brief, periods of idleness.

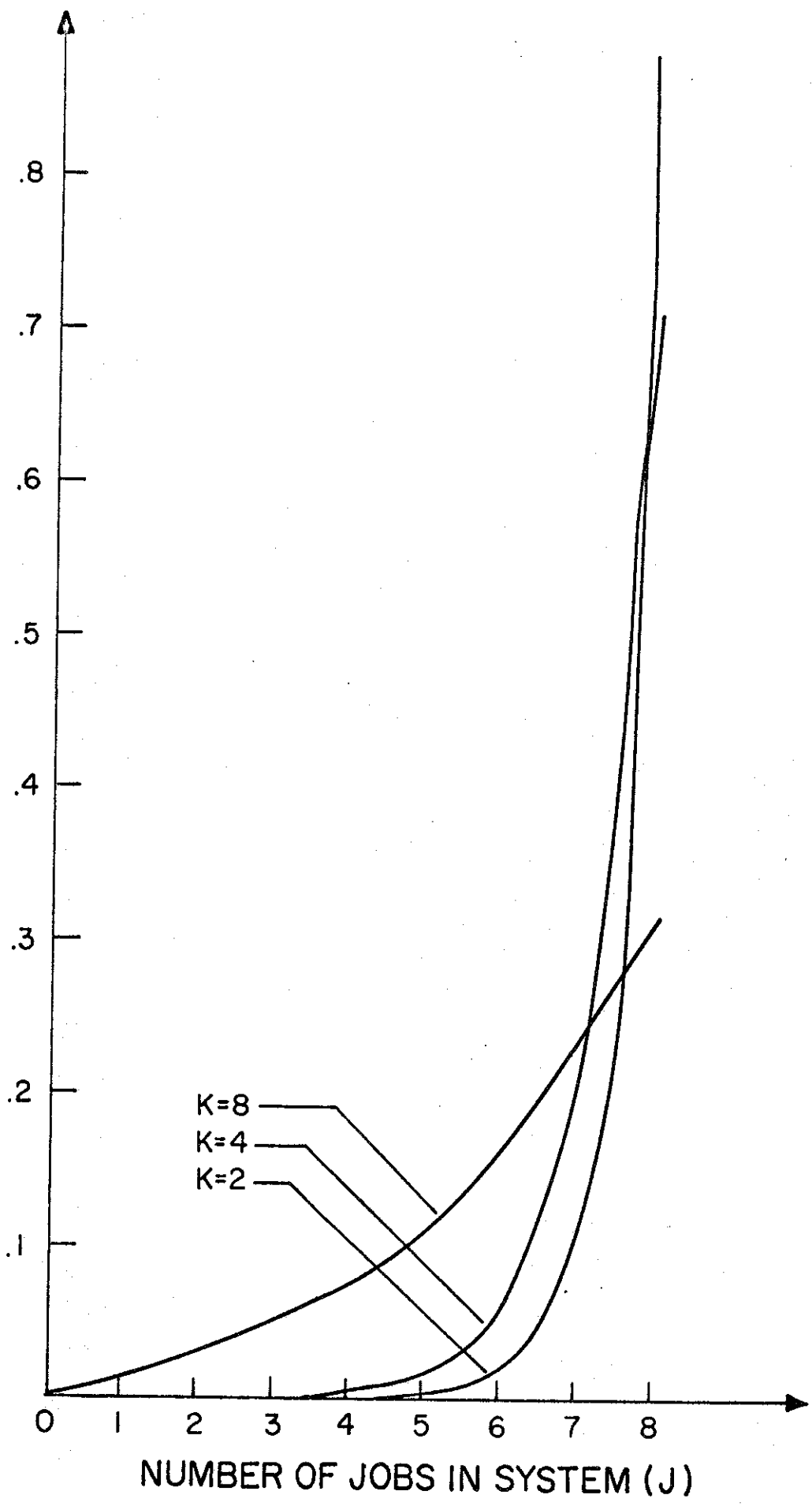
Further evidence of the variability effect is reflected in the plot of arbitrary time state probability values in Figure 2. The curves for the first and second cases appear quite similar, but the third case takes a much different appearance. The availability of unused processing capability (states 0 and 1), although small, is evident in the third case but not in either the first or second.

Figure 3 presents the arbitrary time state probability values for a two processor system with deterministic inter-arrival times of 80, 40, 20, 15 and 10. The shifts in the curves are expected, but the swift change marking the different behavior for 20, 15 and 10 clearly indicates that the saturation point for the system is encountered within this range of values.

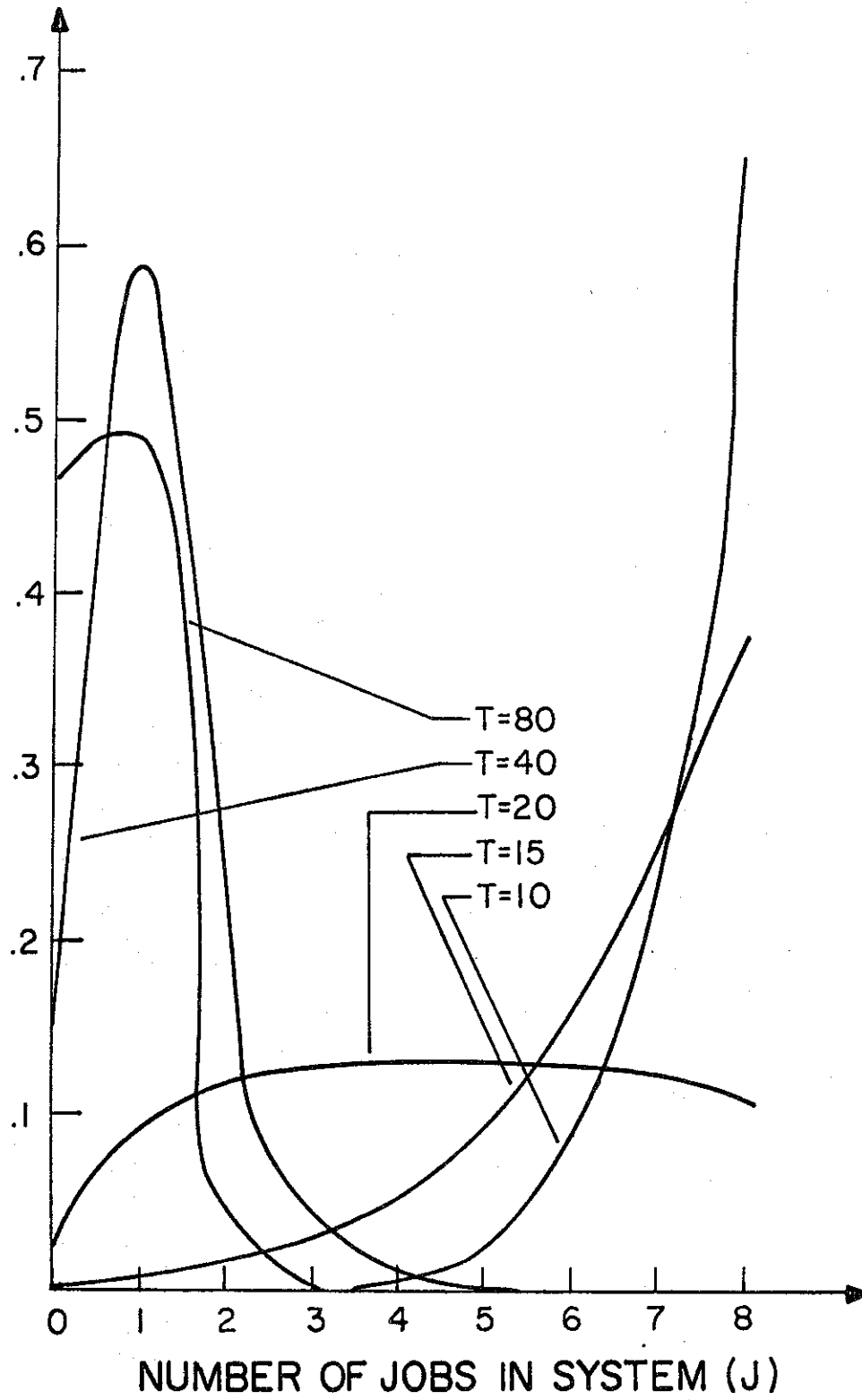
To test the comparative behavior for a system with more, but less capable processors, we describe a system with four identical processors, each having one-half the service rate of the original two processors. The results are presented in Figure 4. With a low demand the resulting behaviors are qualitatively similar but quantitatively rather different. Evidently the lower processing rate is keeping jobs in the 4-processor system longer and the close similarity of the 2-processor curve for  $T=40$  to the 4-processor curve for  $T=80$  suggests that perhaps the 2-processor system is preferable, i.e. it provides roughly analogous behavior under a heavier demand. The values for the blocking probability and processor utilization shown in Table 2 also contribute to suggesting a 2-processor system as preferable. However, two significant facts have not been considered:

- (1) the buffer usage with the 4-processor system is considerably less under higher demand, giving clear indication that a lower buffer capacity could be used in a 4-processor system; and

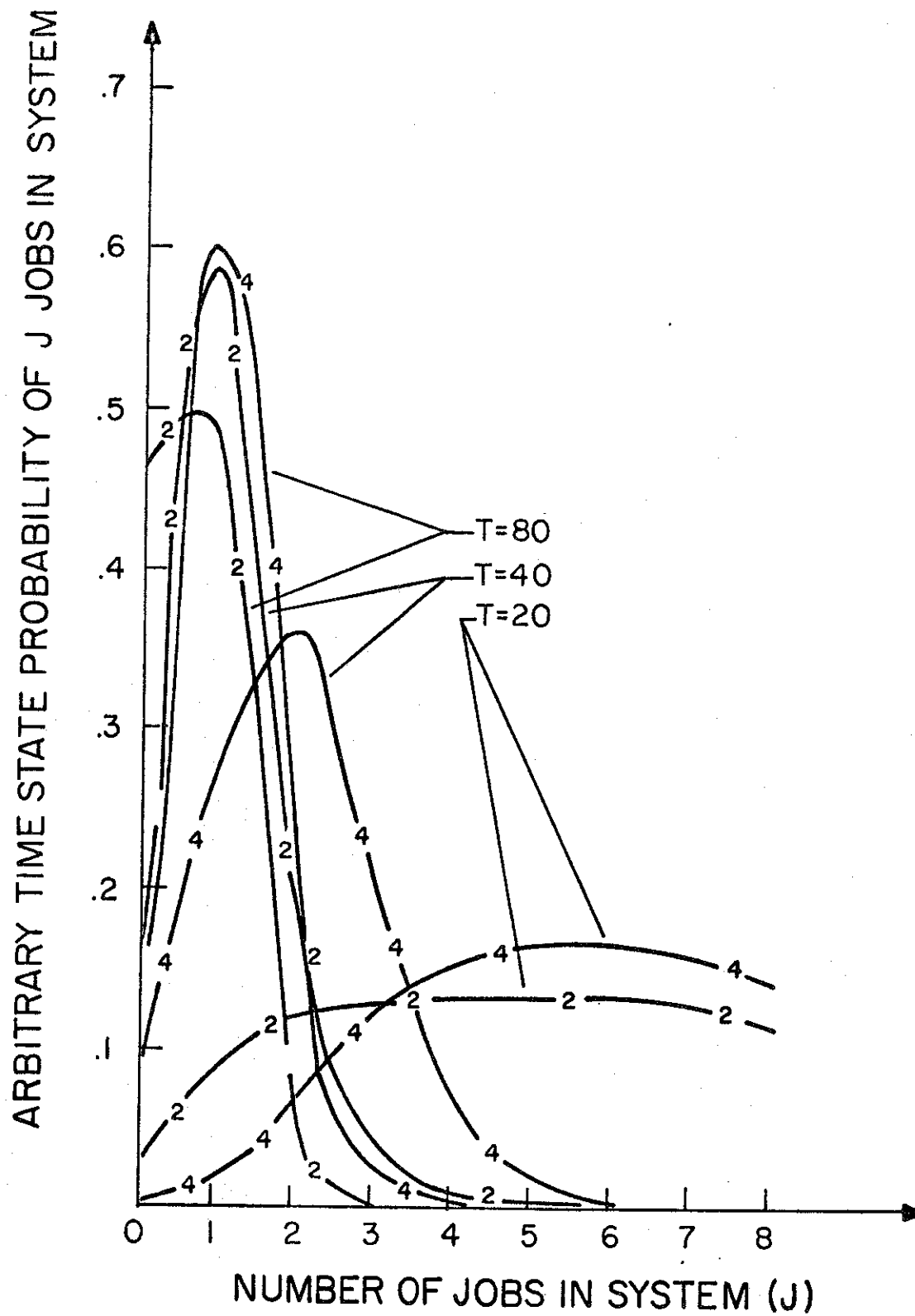
ARBITRARY TIME STATE PROBABILITY OF J JOBS IN SYSTEM



ARBITRARY TIME STATE PROBABILITY OF J JOBS IN SYSTEM







Deterministic Inter-arrival Time	Offered Load per Processor	Probability of Blocking	E(BC) Expected Busy Cycle	Avg.No.Jobs in Input Buffer	PUB Processor Utilization System Busy	PU Processor Utilization
80	s=2	0	94.54	0	.54	.25
	s=4	0	158.60	0	.33	.25
45	s=2	0	79.50	.03	.62	.44
	s=4	.06	245.46	.01	.47	.42
40	s=2	0	81.49	.06	.65	.50
	s=4	.05	298.36	.02	.52	.47
35	s=2	0	87.37	.13	.69	.57
	s=4	.05	396.05	.06	.58	.54
20	s=2	.08	437.49	2.60	.95	.92
	s=4	.09	4803.85	1.51	.92	.91
15	s=2	.25	5151.21	4.62	1.00	1.00
	s=4	.26	54967.63	2.76	.99	.99
10	s=2	.50	320537.69	5.50	1.00	1.00
	s=4	.50	516017.75	3.51	1.00	1.00

$\mu = .025$   $N-s = 6$   $s=2$

$\mu = .0125$   $N-s = 4$   $s=4$

Table 2. Comparative Behavior of Two- and Four-Processor Systems under Increasing Demand.

- (2) most importantly, the cost differential for the less capable processors comprising the 4-processor system could exceed the factor of 2 by a considerable amount.

However, we recognize that a 4-processor system introduces added overhead, not considered in the model. All considered, we must conclude that a general advantage for the 2-processor system cannot be based on the derived behavior.

Note the directional shifts in the expected busy cycle for the 2-processor system ( $T = 80, 45, 40$ ), which is not demonstrated by the 4-processor system. We suspect that this difference in behavior stems from the longer idle periods under low demand for the 2-processor system. As the demand increases, the idle period exceeds the increase in the busy period for a brief time. Also, note the tremendous increase in the busy cycle for the 2-processor system as the inter-arrival time goes from 15 to 10. The magnitude of this jump suggests that a 4-processor system might be more capable of adjusting to increasing demand. We hesitate to offer this conclusion without further study.

As a final point, we remind the reader that the model is developed from the operator perspective. No measures reflecting the user perspective, e.g. response time, are included as behavioral variables. A complete evaluation treatment would include both cost figures and behavioral variables reflecting the user perspective.

#### SUMMARY AND CONCLUSIONS

We have developed a detailed model of processor utilization in a homogeneous multiprocessor computer system. The model assumes a general input process and an exponential processing time for each processor. A finite capacity input buffer is used when no processor is available. The modeling approach derives the arbitrary time state probability distribution, which cannot be determined

using simpler methods. The expected busy cycle follows directly from the arbitrary time state probability distribution, and two measures of processor utilization are obtained. A computational model, requiring several approximations, is developed from the mathematical model. Experimentation with varying input distributions leads to the following conclusions:

- (1) A highly variable input process, i.e. an inter-arrival time distribution with a high variance, causes extremely long busy cycles with the effect exceeding the proportional increase in the average demand.
- (2) Strictly from the operator perspective and neglecting processor costs and buffer usage, a system with two processors, each having twice the processing rate of single processors in a 4-processor system, gives preferable behavior.

## REFERENCES

1. Sayers, Anthony P. Operating Systems Survey, Auerbach Publishers, Inc., 1971.
2. Critchlow, A. J. "Generalized Multiprocessing and Multiprogramming Systems" Proceedings of the FJCC, 1963, 107-126.
3. Kafura, D. G. and V. Y. Shen. "Scheduling Independent Processors with Different Storage Capacities", Proceedings of the ACM, November 1974, 161-166.
4. Graham, R. L. "Bounds on Multiprocessing Timing Anomalies", SIAM J. Applied Mathematics, 17(2): March 1969, 416-129.
5. Muntz, R. R. and E. G. Coffman, Jr. "Preemptive Scheduling of Real Time Tasks on Multiprocessor Systems", J. ACM, 17(2): April 1970, 324-338.
6. Ramamoorthy, C. V., K. M. Chandy and M. J. Gonzalez. "Optimal Scheduling Strategies in a Multiprocessor System", IEEE Transactions on Computers, C-21: February 1972, 137-146.
7. Bovet, D. P. and G. Estrin. "A Dynamic Memory Allocation Algorithm", IEEE Transactions on Computers, C-19: May 1970, 403-411.
8. Rosenfeld, J. L. "A Case Study in Programming for Parallel Processors", Communications of the ACM, 12(12): December 1969, 645-655.
9. Baer, J. L. "A Survey of Some Theoretical Aspects of Multiprocessing", Computing Surveys, 5(1): March 1973, 31-80.
10. Coffman, E. G., Jr. "Bounds on Parallel-Processing of Queues with Multiple-Phase Jobs", Naval Research Logistics Quarterly, 14(3): September 1967, 345-366.
11. Ramamoorthy, C. V. and K. H. Kim. "Pipelining - The Generalized Concept and Sequencing Strategies", Proceedings of the NCC 1974, 43, 289-297.
12. Mitchell, John, Charles Knadler, Gary Lunsford and Steve Yang. "Multiprocessor Performance Analysis", Proceedings of the NCC, 43, May 1974, 399-403.
13. Bhat, U. Narayan and Richard E. Nance. "Busy Period Analysis of a Time-Sharing System Modeled as a Semi-Markov Process", J. ACM, 18(2): April 1971, 221-238.
14. Takacs, Lajos. Introduction to the Theory of Queues, Oxford University Press: New York, 1962.
15. Chen, Y. E. and D. L. Epley. "Memory Requirements in a Multi-processing Environment", J. ACM, 19(1): January 1972, 57-69.
16. Enslow, Philip H., Jr., (Ed). Multiprocessors and Parallel Processing, John Wiley & Sons, Inc.: New York, 1974.

17. Kemeny, J. G. and J. L. Snell. Finite Markov Chains, D. Van Nostrand: Princeton, 1960.
18. Ramamoorthy, C. V. and H. F. Li. "Efficiency in Generalized Pipeline Networks", Proceedings of the NCC 1974, 43, 625-635.