Sequential Processing Machines (S.P.M.) Analyzed With a Queuing Theory Model

LEONARD KLEINROCK

University of California, Los Angeles, California

Abstract. Results are obtained for a model of many processors operating in series. The results are obtained directly by recognizing that a sequential processing may be viewed as a cyclic queue. Exact results are given for two sequential processing stages with a buffer storage of arbitrary size between the stages, and approximate results for the case of 2^M (M an integer) stages. The analysis is good only for exponentially distributed computation times.

Introduction

It is clear that one can always place requirements on the speed of computation which exceed the processing rate for any existing computer. If it is necessary to process at such excessive rates, some form of parallel processing must be used. Whereas many systems of parallel computation have been proposed, they may all be reduced to a canonical structure consisting of two or more processors coupled through a common supervisory control, with each having access to all, or part, of the available memory, as shown in Figure 1. Each of the P processors is considered to be identical in structure, but they may differ in computational speed.[1]

In such a system, the question as to how one should divide up the total computation load presents itself. Indeed, the problem, in general, does not break up into P independent sections (each of which could then be assigned to one of the P processors). It is perhaps more reasonable to assume that the computations taking place in the various processors are interdependent. In particular, it is assumed that the data must be handled by each processor in sequence; that is, the first processor carries out a portion of the total computation on a subset of the data and then transfers these results to the second processor (via a storage area in memory which is accessible to both); the second processor then performs its share of work and passes its results on to the third processor (via memory), and so on. After the first processor finishes its task on the first subset of data, it immediately begins work on the second subset, and so on. Consequently, one may visualize many subsets (or parcels) of data traveling down the series chain of processors and undergoing various stages of processing along the way. We refer to such a chain of Sequential Processing Machines as an S.P.M. system. It is clear that there are two limiting factors here: the computation rate of the processors (possibly all different), and the size of the commonly accessible storage between each processor. Indeed, when any interprocessor storage fills up, then the processor feeding that storage must stop until the succeeding processor empties one memory slot (equal to the room for the results

Preparation of this paper was sponsored in part by the Office of Naval Research and the U. S. Atomic Energy Commission. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

[1] The assumption of identically structured processors is commonly used. However, there is a group at U.C.L.A. which is considering the advantages of allowing the structure of processors to vary with each problem. (See [2, 3].)

Journal of the Association for Computing Machinery, Vol. 13, No. 2 (April, 1966), pp. 179–193

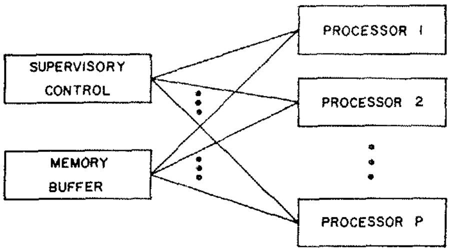

FIG. 1. Parallel processing system

of computation on one subset of data). Similarly, whenever an interprocessor storage empties completely, then the processor which draws data from that storage cannot begin new work until more data become available. The time required to completely process the entire set of data will clearly depend upon these two factors (speed and storage size), but will also depend upon the distribution of computation requirements placed on each processor by each subset of data. (This is discussed in greater detail below.) The expected value of the time to completely process the data is of prime interest, and it is this quantity which we solve for (in certain restricted cases).

Recently, Aoki, Estrin and Mandell [1] considered the analysis of such S.P.M. systems. They obtained results for the case of two processors ($P = 2$), where the distribution of computation time in each processor was completely arbitrary. However, they analyzed the cases in which the size (N) of the buffer storage between processor 1 and processor 2 was restricted to $N = 1$ and $N = \infty$. In this paper, by recognizing that such a system of two series processors is identical to a class of cyclic queues, studied by Koenigsberg [4], we are able to extend the earlier results [1] to the case of a buffer of arbitrary size N between the two stages; however, in so doing, we must restrict the distribution of computation time to be exponential. Furthermore, by approximating the behavior of a system of $P = 2^M$ processors by two processors (each consisting of 2^{M-1} stages), we are able to give approximate results for the multiprocessor case.

The Model

Two of the S.P.M. models considered in [1] may be described as follows. The computation requirement may be thought of as a sequence of n customers, each of which must pass through two processing stations in series; at each of the two stations, a computation is performed for each customer. The time required for each such computation is considered to be a random variable. The processing time t at the first station is assumed to be drawn from an arbitrary but continuous distribution $P_1(t)$, and from $P_2(s)$ for the processing time s at the second station. All service (processing) times are chosen independently, one from the other. The reason for viewing these processing times as random variables is that the particular numbers involved in each calculation affect instruction execution times as well as the number of iterations around loops of instructions. Since in general we cannot predict the input data that are to be used, we choose to consider the time required to perform calculations on these data as random variables.

Each customer must be fully processed by the first facility before he is allowed to enter the second facility. Let N be defined as number of customers which the system can "store" or provide waiting room for between the completion of their service at the first station and before the initiation of their service at the second station; i.e., the maximum allowable queue that may form in front of station 2 is equal to N. This corresponds to the size of the common memory space in the memory buffer expressed in units of number of available slots for each customer's partially completed computation. Aoki, Estrin and Mandell [1] consider two special cases, namely $N = 1$ and $N = \infty$. In the first case, this corresponds to requiring that the first processor cannot begin service on the $(i+1)$-st customer until the second processor completes the service of the ith customer, since the ith customer is occupying all of the available space ($N = 1$) on the queue. If this rule is violated, it can be seen that the ith customer's partially completed results will be wiped out as the results for the $(i+1)$-st customer are stored into the same memory space. In the second case ($N = \infty$), no restriction is placed on the first processor since there is always room in memory for storage of its results. The performance criterion for such a system is taken to be proportional to the expected time, T', to complete the processing for all n customers (each of which must pass through both facilities). It is assumed that $n \gg N$. We refer to the model described above as the *computer model* of an S.P.M. system.

We now demonstrate that the computer model (for any value of N) is equivalent to a cyclic queue [4]. Consider again two service facilities in tandem with a maximum allowable queue of size $N-1$ in front of *each* facility.[2] Prime the system by placing N customers at the first facility ($N-1$ on queue and one in service). When a customer leaves station 2 he immediately joins the queue in front of station 1 and again goes through the two series facilities (hence the name cyclic). (See Figure 2.) The service time t at the first station is chosen from a continuous distribution $P_1(t)$ and the service time s, at the second facility, from $P_2(s)$. Observe that the second service facility cannot begin service on a customer until he has been serviced by the first facility. In addition, when the second queue is full, there are no customers available for the first service facility, since all N customers are at the second station. The key point is that a queuing theory model of an S.P.M. system has been set up in which the "blocking" phenomenon of the computer model described earlier is brought about in this new model by limitation of the arrival of customers to the first service facility. The only time that a customer may enter the first facility is when there is "room" for such a customer in the second facility. Thus there is an arrival pattern at the first facility that is dependent on the queue size of the second facility[3]; this is exactly the situation we need to describe the blocking condition. The fact that the same "customers" are going around the system many times is of no concern, since their service (computation) time is chosen independently at each step. If these N customers are sent around the system n/N times, then each service facility will have serviced n customers. Consequently, the expected time, T, to complete the n/N cycles in this *queuing model* will be the same as the expected time, T', to process n customers in the computer model.

[2] The queue size is $N-1$ instead of N, since in our queuing model it is assumed that there is room for one customer in the service facility itself; in the computer model it is assumed that there is no such storage capacity in the processor itself.

[3] Due to the symmetry of the two service facilities in this model, all comments referring to the first station also apply to the second station. The only exception is in the initial configuration, where we insist that the N units begin at the first station.

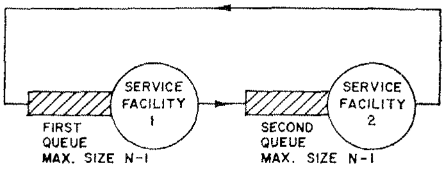

FIG. 2. The cyclic queue model

Bounds on Performance

Following the suggestion by Aoki, Estrin and Mandell, we define the performance measure, R, as the ratio of T' (the expected time to process the n customers in the multiprocessor system—see Figure 1) to the expected time that it would take a single processor to serve these n customers by itself. The average time, m^*, to service one customer in this single (reference) processor will be taken as the average of the mean computation times of the P processors in the multiprocessor system; that is, if m_p denotes the average time that the pth processor takes to process one customer, then

$$m^* = \frac{1}{P} \sum_{p=1}^{P} m_p \tag{1}$$

Now, since there is a total of n customers passing through P processing stations, it is clear that the single processor system must service each customer P times in order to perform the same total computation. Since each computation time is an independent random variable by assumption, this single processor must in effect carry out nP computations (each taking on the average, m^* seconds). Consequently,

$$R = \frac{T'}{nPm^*} \tag{2}$$

Let us consider the range of values which R may take. It is clear that the multiprocessor system will always perform better than the case in which only one of its P processors is actively computing at any given time. In this degraded situation, the expected time to completely process n customers will merely be the sum of the average time that it takes each processor to service n customers (i.e., $T' = n \sum_{p=1}^{P} m_p$). Consequently,

$$R \le \frac{n \sum_{p=1}^{P} m_p}{nPm^*} = 1.$$

Furthermore, the best that can be obtained is a situation in which all P processors are actively computing all of the time. In this best case, T' must be greater than or equal to the time that it takes the slowest of the P processors to service n customers (i.e., $T' \ge nm_{\max}$). But $m_{\max} \ge m^*$, and so

$$R \ge \frac{nm^*}{nPm^*} = \frac{1}{P}.$$

Thus, for any S.P.M. system with P processors, R is bounded from above and below by

$$\frac{1}{P} \le R \le 1. \tag{3}$$

Obviously, the smaller is R, the better is the performance of the S.P.M. system. Below, we consider only the case for which the distribution of computation time, t, in the pth processor is given by

$$P_p(t) = 1 - e^{-\mu_p t}, \tag{4}$$

i.e., $P_p(t)$ is the probability that the computation time in the pth processor is less than or equal to t. From this it is clear that $m_p = 1/\mu_p$, and so

$$m^* = \frac{1}{P} \sum_{p=1}^{P} 1/\mu_p .$$

Two-Processor Systems

The first result requires that the value of P be restricted to $P = 2$. It is convenient to define the quantity

$$\sigma = \frac{m_1}{m_2} = \frac{\mu_2}{\mu_1}.$$

THEOREM 1. *Consider a two-processor S.P.M. system described by the computer model above in which the distribution of computation times is exponential as given in Equation* (4) *and which provides storage for N customers between the two stages. The performance measure R (for $n \gg N$) is*

$$R = \begin{cases} \dfrac{1}{1+\sigma}\,\dfrac{1-\sigma^{N+1}}{1-\sigma^N} & \sigma \neq 1, \\[2ex] \dfrac{N+1}{2N} & \sigma = 1. \end{cases} \tag{5}$$

PROOF. As shown above, the value of T calculated using the queueing model must be equal to T' as calculated using the computer model. Consequently, the theorem will be proved by solving for T.

To review, the cyclic queueing model, as shown in Figure 2, consists of N customers distributed between the two processors. In 1958 Koenigsberg [4] considered the problem of cyclic queues with exponentially distributed service times (see Equation (4)), and solved for the equilibrium probability $p(k, N-k)$ of finding k customers (waiting and in service) at the first facility and $N-k$ customers at the second facility. His result (for $P = 2$) is

$$p(k, N-k) = \sigma^{k-N} p(N, 0) \qquad (k = 0, 1, 2, \cdots, N) \tag{6}$$

where by definition $\sigma = \mu_2/\mu_1$. The value of $p(N, 0)$ may be found by summing Equation (6) over all k and equating this sum to unity, i.e.,

$$\sum_{k=0}^{N} p(k, N-k) = 1 = \sum_{k=0}^{N} \sigma^{k-N} p(N, 0).$$

Thus,

$$p(N, 0) = \frac{\sigma^N}{\sum_{k=0}^{N} \sigma^k}.$$

Consequently,

$$p(N, 0) = \begin{cases} \sigma^N \dfrac{1 - \sigma}{1 - \sigma^{N+1}} & \sigma \neq 1, \\ \dfrac{1}{N+1} & \sigma = 1. \end{cases}$$

Thus, Equation (6) becomes (for $k = 0, 1, 2, \cdots, N$),

$$p(k, N-k) = \begin{cases} \dfrac{1 - \sigma}{1 - \sigma^{N+1}} \sigma^k & \sigma \neq 1, \\ \dfrac{1}{N+1} & \sigma = 1. \end{cases} \tag{7}$$

If we make use of this equilibrium (steady-state) probability distribution, we require that the system has been operating for a long time. This is equivalent to requiring that each of the N customers go around the cycle many times. Now, since n/N is just the number of cycles needed to process a total of n customers, we insist that $n/N \gg 1$, or $n \gg N$.

For $p = 1, 2$, E_p is defined as the probability that the pth facility is empty. It is clear that

$$E_1 = p(0, N) \tag{8}$$

and

$$E_2 = p(N, 0). \tag{9}$$

Both expressions are readily evaluated from Equation (7). From Equation (2) we get (for $P = 2$),

$$R = \frac{T'}{2nm^*} = \frac{T}{2n \dfrac{m_1 + m_2}{2}} = \frac{T}{n(m_1 + m_2)}.$$

But, since $m_p = 1/\mu_p$,

$$R = \frac{T}{n\left(\dfrac{1}{\mu_1} + \dfrac{1}{\mu_2}\right)}. \tag{10}$$

Now, since $T(= T')$ represents the expected time that it takes to completely process the n customers, it is found that the expected time during which the pth facility is busy is merely $(1 - E_p)T$. During these $(1 - E_p)T$ seconds, the pth facility must service n customers, each of which takes an average of $1/\mu_p$ seconds. Consequently,

$$(1 - E_p)T = n/\mu_p \qquad (p = 1, 2).$$

Summing this last equation over p, we obtain

$$(2 - E_1 - E_2)T = n\left(\frac{1}{\mu_1} + \frac{1}{\mu_2}\right).$$

From this last and Equation (10) it can be seen that

$$R = \frac{1}{2 - E_1 - E_2}. \tag{11}$$

This denominator is now evaluated by use of Equations (7), (8) and (9). For $\sigma \neq 1$, we obtain

$$\begin{aligned} 2 - E_1 - E_2 &= 2 - \frac{1-\sigma}{1-\sigma^{N+1}} - \sigma^N \frac{1-\sigma}{1-\sigma^{N+1}} \\ &= 1 + \frac{1 - \sigma^{N+1} - (1-\sigma)(1+\sigma^N)}{1-\sigma^{N+1}}. \end{aligned}$$

Thus,

$$2 - E_1 - E_2 = 1 + \frac{\sigma - \sigma^N}{1 - \sigma^{N+1}} \qquad \sigma \neq 1. \tag{12}$$

For $\sigma = 1$, we obtain

$$2 - E_1 - E_2 = 2\left(1 - \frac{1}{N+1}\right)$$

or

$$2 - E_1 - E_2 = \frac{2N}{N+1} \qquad \sigma = 1. \tag{13}$$

Combining Equations (11), (12) and (13), we finally arrive at

$$R = \begin{cases} \dfrac{1}{1+\sigma}\dfrac{1-\sigma^{N+1}}{1-\sigma^N} & \sigma \neq 1, \\[2ex] \dfrac{N+1}{2N} & \sigma = 1, \end{cases}$$

which completes the proof of the theorem.

Note that $R = R(\sigma, N)$ is symmetrical in σ and σ^{-1}; that is,

$$R(\sigma, N) = R(\sigma^{-1}, N).$$

This is easily shown as follows. For $\sigma = 1$, it is clearly true. For $\sigma \neq 1$,

$$\begin{aligned} R(\sigma, N) &= \frac{1}{1+\sigma}\left(\frac{\sigma^{-1}}{\sigma^{-1}}\right)\frac{1-\sigma^{N+1}}{1-\sigma^N}\left(\frac{\sigma^{-N}}{\sigma^{-N}}\right) \\ &= \frac{1}{1+\sigma^{-1}}\frac{1-\sigma^{-(N+1)}}{1-\sigma^{-N}} \\ &= R(\sigma^{-1}, N). \end{aligned}$$

$R(\sigma, N)$ is plotted in Figure 3.

For $\sigma = 0, \infty$, it can be seen that $R(0, N) = 1$ independent of N. Physically, this makes sense since one or the other of the two computation facilities is infinitely slower than the other; comparing this to a system with one processor whose average computation time is the arithmetic mean of those two must result in a system which gives an average total computation time equal to that of the two-processor system.

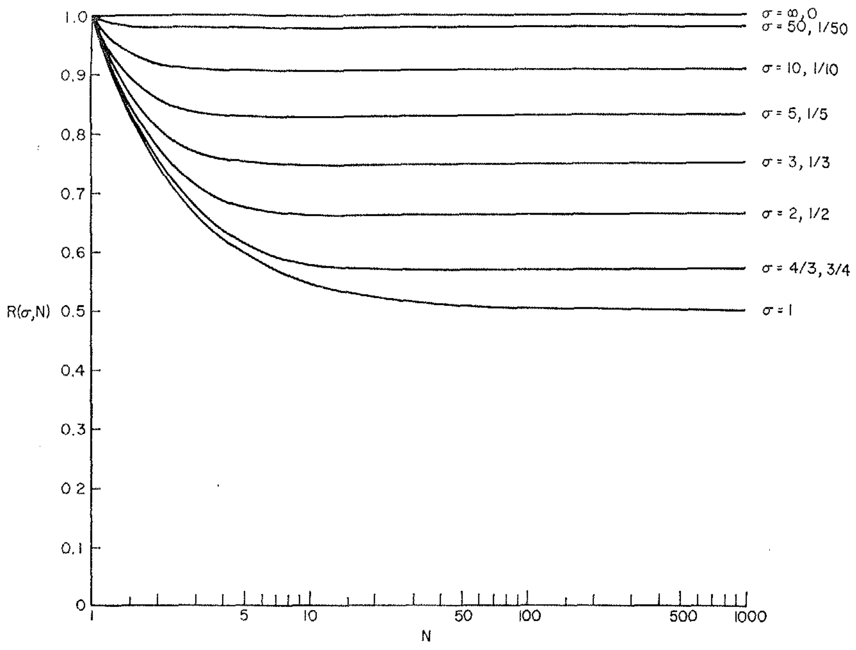

FIG. 3. Performance measure $R(\sigma, N)$

From Figure 3, it can be seen that $R(\sigma, N)$ seems to be a monotonically decreasing function of σ. This is shown to be true in the following theorem.

THEOREM 2. *For*[4] $\sigma_1 \leq \sigma_2 \leq 1, \quad R(\sigma_1, N) \geq R(\sigma_2, N)$.

PROOF. From Theorem 1, we have (for $\sigma \neq 1$)

$$R(\sigma, N) = \frac{1 - \sigma^{N+1}}{1 - \sigma^{N+1} + \sigma - \sigma^N} = \frac{1}{1 + \dfrac{\sigma - \sigma^N}{1 - \sigma^{N+1}}}.$$

Let

$$x(\sigma) = \frac{\sigma - \sigma^N}{1 - \sigma^{N+1}}.$$

It is enough to show that $x(\sigma)$ is monotonically increasing with σ (for $0 \leq \sigma < 1$). This is done by showing that $dx/d\sigma \geq 0$. Differentiating, we get

$$\frac{dx(\sigma)}{d\sigma} = \frac{1 - N\sigma^{N-1} + N\sigma^{N+1} - \sigma^{2N}}{(1 - \sigma^{N+1})^2}. \tag{14}$$

The denominator in Equation (14) is always positive (for $0 \leq \sigma < 1$), so it may be neglected. Denoting the numerator by $y(\sigma)$, we have

$$\begin{aligned} y(\sigma) &= 1 - N\sigma^{N-1} + N\sigma^{N+1} - \sigma^{2N} \\ &= (1 - \sigma^{2N}) - N\sigma^{N-1}(1 - \sigma^2) \\ &= (1 - \sigma^2)[1 + \sigma^2 + \sigma^4 + \cdots + \sigma^{2N-2} - N\sigma^{N-1}]. \end{aligned}$$

[4] Restricting $\sigma_1, \sigma_2 \leq 1$ is completely general, since $R(\sigma, N) = R(\sigma^{-1}, N)$.

Again, since $1 - \sigma^2$ is always positive, it is necessary to consider only the polynomial $z(\sigma) = y(\sigma)/1 - \sigma^2$ whose degree is $2N-2$.

$$z(\sigma) = 1 + \sigma^2 + \sigma^4 + \cdots + \sigma^{2N-2} - N\sigma^{N-1}. \tag{15}$$

By Descartes' rule of signs for polynomials, it can be seen that Equation (15) has at most two positive real roots. Clearly, one of these is at $\sigma = 1$. Also, it can be seen that

$$z(\sigma) = \sigma^{2N-2} z(\sigma^{-1}).$$

Therefore, if σ_0 is a root of $z(\sigma)$, so must be σ_0^{-1}. Thus, the only two positive real roots of $z(\sigma)$ must *both* occur at $\sigma = 1$. Furthermore, $z(0) = 1 > 0$. Thus,

$$z(\sigma) > 0 \quad \text{for} \quad 0 \leq \sigma < 1$$

Consequently, $dx/d\sigma > 0$ for $0 \leq \sigma < 1$, which completes the proof of Theorem 2.

From Theorem 1, we have $R(\sigma, 1) = 1$ for all σ. This last, with Theorem 2, gives the following Corollary.

COROLLARY.

$$R(1, N) < R(\sigma, N) \quad \textit{for} \quad \sigma \neq 1,\ N > 1. \tag{16}$$

Note further, from Theorem 1, that

$$\lim_{N\to\infty} R(\sigma, N) = \begin{cases} \dfrac{1}{1+\sigma} & \sigma \leq 1, \\ \dfrac{1}{1+\sigma^{-1}} & \sigma \geq 1. \end{cases} \tag{17}$$

The fact that $\sigma = 1$ is optimum (Corollary above) is reasonable from a physical point of view, since both facilities are truly symmetrical (as can be seen from the cyclic queueing model); consequently, if they do not both have identical computational speeds, the faster facility will, on the average, be waiting for the slower facility. The balanced case ($\sigma = 1$) is clearly the best arrangement, as one would expect by intuition.

It is interesting to note that a relatively small storage capability (N in the range from 10 to 20) is sufficient to achieve a performance measure which is within a few percent of its optimum (minimum) value (see Figure 3).

Multiprocessor Systems

The analysis is now extended to S.P.M. systems for which $P > 2$. Unfortunately, the cyclic queueing model proposed earlier no longer gives an accurate description of such systems. The source of the difficulty is that we cannot cause the appropriate blocking phenomena in the queueing model, since customers may now distribute themselves among many processing stations rather than among just two stations. Consequently, we resort to an approximation which is based on our results for the $P = 2$ case.[5]

The subscript P is now introduced into the notation for the performance measure

[5] A storage of size N between each processor is assumed.

R and processing time T of a P-processor system; that is,

$$R = R_P$$
$$T = T_P.$$

The average time necessary to process a single customer in a P-processor S.P.M. system is just T_P/n. Equations (2) and (3) then become

$$R_P = \frac{T_P}{nPm^*} \tag{18}$$

and

$$\frac{1}{P} \le R_P \le 1. \tag{19}$$

Consider $P = 4$. Having already analyzed the behavior of the $P = 2$ systems, we are led to consider the configuration shown in Figure 4, in which the $P = 4$ system is considered as two $P = 2$ systems in tandem. We are tempted to replace each $P = 2$ system by a single $P = 1$ system whose average processing time is just T_2/n. Were this to be done, we would be left with a simple $P = 2$ system, whose behavior has already been calculated. However, once this is done, the assumption of exponentially distributed service times (Equation (4)) is violated, since we now have a two-stage service operation at each processing stage. We choose to make this replacement, however, and thereby only approximate the true system behavior. The results below, under this approximation, indicate that the approximation is rather good. In this way, (approximate) answers for $P = 4$ are obtained.

Now consider $P = 8$ and break this system into two $P = 4$ systems. Then use the results from Theorem 1 with T_4/n as the average time to pass through each $P = 4$ system. Continuing, consider $P = 2^M$ (M an integer) and decompose such a system into two $P = 2^{M-1}$ systems, as shown in Figure 5.

This approximate analysis is carried out for two special cases, namely:

(a) All $m_p = m$ (i.e., $1/\mu_p = 1/\mu$) and arbitrary N,

(b) $N \to \infty$, and arbitrary $m_p = 1/\mu_p$.

As we shall see, one or the other of these usually gives a good approximation to the more general cases.

For arbitrary N, and for $1/\mu_p = 1/\mu$ for all $p = 1, 2, \cdots, P = 2^M$, we have from Equation (18)

$$R_{2^M} = \frac{T_{2^M}}{2^M nm^*}. \tag{20}$$

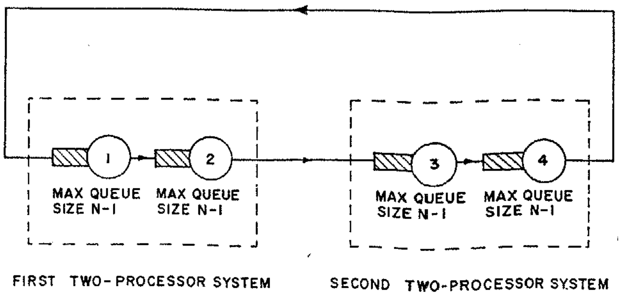

FIG. 4. Decomposition of a four-processor system into two two-processor systems

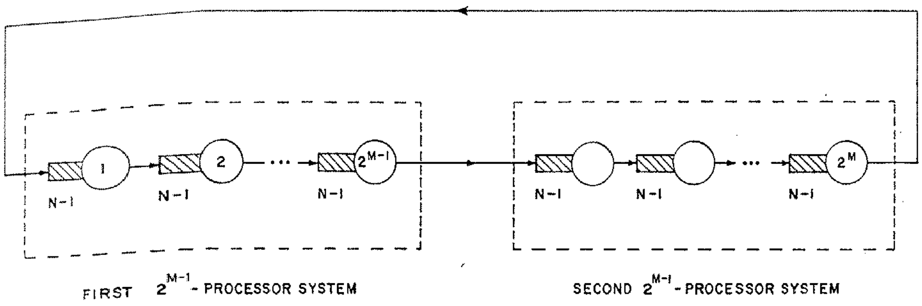
1
2
2^{M-1}
2^M
N-1
FIRST 2^{M-1} - PROCESSOR SYSTEM
SECOND 2^{M-1} - PROCESSOR SYSTEM

FIG. 5. Decomposition of a 2^M-processor S.P.M. system into two 2^{M-1}-processor S.P.M. systems

But $m^* = 1/\mu$ and, using Equation (20) and our approximation (see Figure 5),

$$T_{2^M} \simeq 2n\left(\frac{T_{2^{M-1}}}{n}\right)R_2 \tag{21}$$

where $T_{2^{M-1}}/n$ is the average time for a customer to pass through a 2^{M-1}-processor system. Continuing, we have

$$\frac{T_{2^{M-1}}}{n} \simeq 2\left(\frac{T_{2^{M-2}}}{n}\right)R_2 . \tag{22}$$

From Equations (21) and (22),

$$\frac{T_{2^M}}{n} \simeq (2R_2)^2\frac{T_{2^{M-2}}}{n} . \tag{23}$$

Continuing this iteration, finally,

$$\frac{T_{2^M}}{n} \simeq (2R_2)^M\frac{T_1}{n} . \tag{24}$$

But T_1/n is just the average time for a unit to pass through a single processor, so $T_1/n = 1/\mu$ by definition. Thus

$$\frac{T_{2^M}}{n} \simeq (2R_2)^M\frac{1}{\mu} . \tag{25}$$

From Equations (20) and (25),

$$R_{2^M} \simeq (R_2)^M \tag{26}$$

Each of the R_2 terms generated in the above analysis apply to two identical 2^m-processor systems, so $\sigma = 1$ for each such term.[6] Consequently, using Theorem 1 in Equation (26), we finally have the following:

THEOREM 3. *For all* $m_p = 1/\mu$ $(p = 1, 2, \cdots, P = 2^M)$,

$$R_{2^M} \simeq \left(\frac{N+1}{2N}\right)^M . \tag{27}$$

[6] That is, for two 2^m-processor systems in tandem, σ refers to the ratio of the average time for a customer to pass through one system to his average time to pass through the other system.

Note from the Corollary to Theorem 2 that the best system is one in which all processors have equal computational speeds. This is the case given in Theorem 3, and we find that such a system compares extremely favorably with the ideal system represented by the lower bound in Equation (19). That is, we must compare $[(N+1)/2N]^M$ and $1/2^M$. It can be seen that for those values of N and M such that $(N+1/N)^M$ is reasonably close to unity, the performance of the balanced system is very nearly as good as that of the ideal system. In Figure 6, these systems are compared for various N and M.

For $N \to \infty$ and arbitrary $m_p = 1/\mu_p$, we have, again,

$$R_{2^M} = \frac{T_{2^M}}{2^M n m^*}. \tag{28}$$

Now,

$$m^* = \frac{1}{2^M} \sum_{p=1}^{2^M} 1/\mu_p \tag{29}$$

and so

$$R_{2^M} = \frac{T_{2^M}}{n \sum_{p=1}^{2^M} 1/\mu_p}. \tag{30}$$

T_{2^M}/n is now formed by working up from small values of M. Let

$\dfrac{T_{2^m}^{(p)}}{n}$ = average time to pass one customer through the pth $(p = 1, 2)$ 2^m-processor stage in a decomposed 2^{m+1}-processor system $(m = 0, 1, \cdots, M-1)$,

$$\frac{T_{2^m}^{(\max)}}{n} = \max\left[\frac{T_{2^m}^{(1)}}{n}, \frac{T_{2^m}^{(2)}}{n}\right],$$

$$\frac{T_{2^m}^{(\min)}}{n} = \min\left[\frac{T_{2^m}^{(1)}}{n}, \frac{T_{2^m}^{(2)}}{n}\right].$$

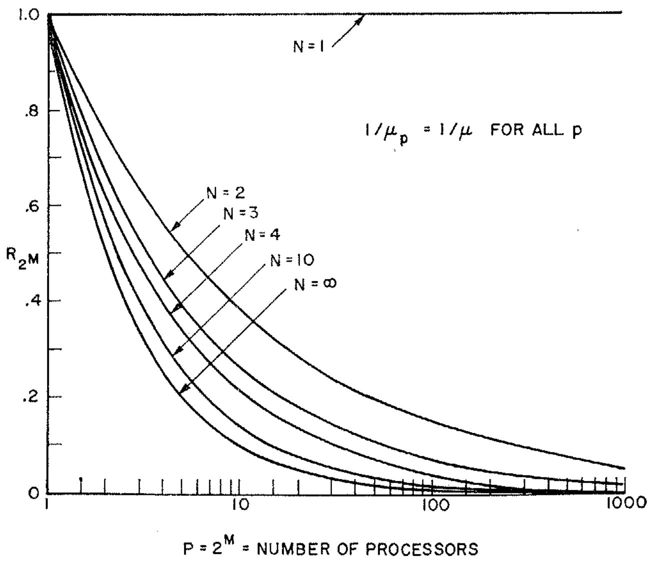

Fig. 6. Performance of balanced S.P.M. system ($1/\mu_p = 1/\mu$) as a function of the number of processors P, with storage (N) as a parameter

For such a 2^{m+1}-processor system, the appropriate definition of σ ($\sigma \leq 1$ has been chosen with no loss of generality) is

$$\sigma = \frac{T_{2^m}^{(\min)}/n}{T_{2^m}^{(\max)}/n} = \frac{T_{2^m}^{(\min)}}{T_{2^m}^{(\max)}}. \tag{31}$$

Beginning with $M = 1$, we have from Equations (17), (28) and (29),

$$\frac{T_2}{n} \simeq 2\left(\frac{1}{2}\right)\left[\frac{T_1^{(\max)}}{n} + \frac{T_1^{(\min)}}{n}\right]\frac{1}{1 + (T_1^{(\min)}/T_1^{(\max)})} = \frac{T_1^{(\max)}}{n}.$$

But for the pth processor, $T_1/n = m_p = 1/\mu_p$, and so

$$\frac{T_2}{n} \simeq \max\left(\frac{1}{\mu_1}, \frac{1}{\mu_2}\right).$$

For $M = 2$,

$$\begin{aligned}\frac{T_4}{n} &\simeq 2\left(\frac{1}{2}\right)\left[\frac{T_2^{(\max)}}{n} + \frac{T_2^{(\min)}}{n}\right]\frac{1}{1 + (T_2^{(\min)}/T_2^{(\max)})}\\ &= \frac{T_2^{(\max)}}{n}\\ &= \max\left[\max\left(\frac{1}{\mu_1}, \frac{1}{\mu_2}\right), \max\left(\frac{1}{\mu_3}, \frac{1}{\mu_4}\right)\right]\\ &= \max\left(\frac{1}{\mu_1}, \frac{1}{\mu_2}, \frac{1}{\mu_3}, \frac{1}{\mu_4}\right).\end{aligned}$$

In general,

$$\begin{aligned}\frac{T_{2^m}}{n} &\simeq \frac{T_{2^{m-1}}^{(\max)}}{n}\\ &= \max\left[\max\left(\frac{1}{\mu_1}, \cdots, \frac{1}{\mu_{2^{m-1}}}\right), \max\left(\frac{1}{\mu_{2^{m-1}+1}}, \cdots, \frac{1}{\mu_{2^m}}\right)\right]\end{aligned}$$

or

$$\frac{T_{2^m}}{n} \simeq \max\left(\frac{1}{\mu_1}, \frac{1}{\mu_2}, \cdots, \frac{1}{\mu_{2^m}}\right). \tag{32}$$

Equations (30) and (32), then, lead to

THEOREM 4. *For $N \to \infty$ and arbitrary $m_p = 1/\mu_p$,*

$$R_{2^M} \simeq \frac{\max\limits_p (1/\mu_p)}{\sum\limits_{p=1}^{2^M} 1/\mu_p}. \tag{33}$$

In Figure 7, this function is plotted versus $P = 2^M$ for various sets of $\{1/\mu_p\}$. It should be noted that for $\mu_p = \mu$, we get $R_{2^M} = 1/2^M$, which is the limiting case of Equation (27) as $N \to \infty$, as of course it must be. It can be seen that Theorem 4 also applies for arbitrary P (not necessarily of the form $P = 2^M$).

For the general case of arbitrary N and m_p, it is necessary to use the more general form

$$R_2 = \frac{1}{1+\sigma}\,\frac{1-\sigma^{N+1}}{1-\sigma^N} \tag{34}$$

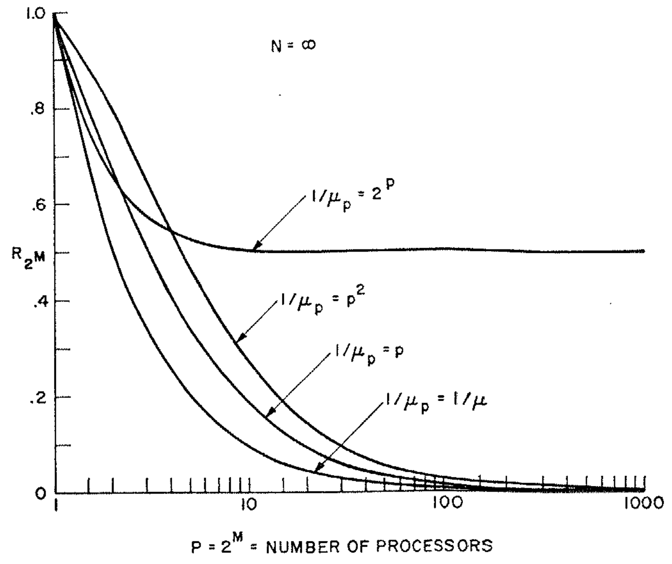

FIG. 7. Performance of an infinite storage ($N \to \infty$) S.P.M. system for various assignments of $1/\mu_p$

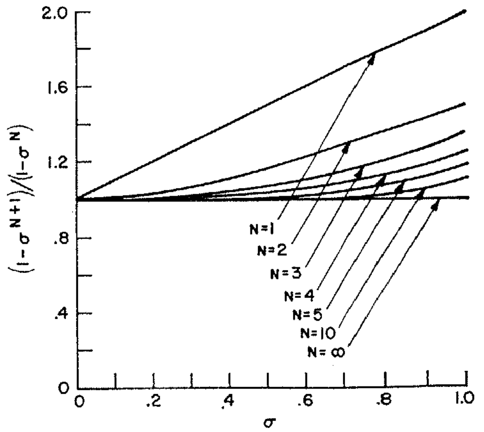

FIG. 8. Effect of approximating equation (34) by $R_2 = 1/1 + \sigma$

in place of Equation (17) (which was appropriate for $N \to \infty$) in Theorem 4 or in place of $(N + 1)/2N$ (which was appropriate for $1/\mu_p = 1/\mu$) in Theorem 3. When N and σ are such that $\sigma^N \ll 1$, then Equation (34) approaches the expression in Equation (17) and then Theorem 4 applies. On the other hand, σ^N will not be much less than unity if σ itself is very close to unity (for reasonably large N, say $N > 20$); in this case, $R_2 \to (N + 1)/2N$ and then Theorem 3 applies. Thus, almost all cases fall into one or the other of the two cases considered (as long as N is not very small). In Figure 8 the factor $(1 - \sigma^{N+1})/(1 - \sigma^N)$ is plotted from Equation (34). Note that this factor goes to $(N + 1)/N$ as $\sigma \to 1$.[7] From Figure 8 it can be seen that the

[7] It is easy to show that $(1 - \sigma^{N+1})/(1 - \sigma^N) \leq (N + 1)/N$ for $0 \leq \sigma \leq 1$, thereby providing an excellent upper bound for this factor.

two cases considered earlier cover most of the interesting possibilities, since the curves tend to cluster around unity.

Conclusions

This paper has been concerned with the behavior of Sequential Processing Machine (S.P.M.) systems. The measure of performance chosen is the ratio R_P of the average time it takes to process n jobs (customers) through a series of P tandem processors to the time it would take a single machine (whose processing time is the average of the processing times for the P machines) to perform the same processing by itself.

Two major results have been presented. The first comes from the recognition that when $P = 2$ the S.P.M. system may be replaced by a cyclic queueing model and thereby obtain exact results for R_2 with an arbitrary storage (of size $N \ll n$) between the processors, with the restriction to exponentially distributed processing times. These results are given in Theorems 1 and 2 (and its corollary) and in Equation (17).

The second major result comes from the approximation of the behavior of a $P = 2^M$ S.P.M. system by a $P = 2$ S.P.M. system, where each processor in this decomposed system consists of 2^{M-1} processors and has an average processing time equal to the solution obtained by approximating a $P = 2^{M-1}$ S.P.M. system by two $P = 2^{M-2}$ sections, and so forth. At each stage of the approximation, the exact results from the $P = 2$ case are used. The main results are given in Theorems 3 and 4. A major conclusion drawn is that the best S.P.M. system is one in which each of the processors has the same average processing time (as one would expect intuitively).

RECEIVED AUGUST, 1964; REVISED AUGUST, 1965

REFERENCES

1. AOKI, M., ESTRIN, G., AND MANDELL, R. A probabilistic analysis of computing load assignment in a multi-processor computing system. Proc. Fall Joint Comput. Conf. 1963, 147–160.
2. ESTRIN, G., BUSSELL, B., TURN, R., AND BIBB, J. Parallel processing in a restructurable computer system. *Trans. IEEE EC-12*, 6 (Dec. 1963), 747–755.
3. ESTRIN, G. AND TURN, R. Automatic assignment of computations in a variable structure computer system. *Trans. IEEE EC-12*, 6 (Dec. 1963), 755–773.
4. KOENIGSBERG, E. Cyclic queues. *Oper. Res. Quart. 9*, 1 (1958).