This paper presents a mathematical model to measure the amount by which a computer's speed is reduced when it time-shares storage with other computers and I/O *channels. The method can be applied to any number of processors and/or channels and storage units, although the complexity of the solution does increase rapidly as the number of processors increases. Explicit formulas and numerical results are given for several special cases.*

The results of a simulation of a shared-memory multiprocessor are presented, showing how closely the mathematical model fits the operation of a simulated system.

Effects of storage contention on system performance

by C. E. Skinner and J. R. Asher

A central processing unit may be required to share main storage units with other central processing units as well as with input/output channels. The effect on speed of a reference processor contending for use of a storage unit can be determined to a reasonable degree of accuracy by use of a *stretching factor*. The time needed to execute a program without contention is multiplied by the stretching factor to determine execution time in the presence of contention.

A mathematical model, consisting of a number of interacting processors and shared main storage units, allows determination of the stretching factor. This paper describes the mathematical model, including the simplifying assumptions made, and considers application of the method to several processor-storage unit combinations. It then compares results with those obtained by simulation of a shared-storage multiprocessor.

The mathematical model

In order to make the mathematical analysis tractable, certain assumptions and simplifications are made. However, a restrictive assumption does not reduce the scope of application. Indeed, one must always be prepared to optimally fit a model to its image.

The following conventions apply in this paper:

- Each storage unit, independently of the others, operates continuously in a cyclic fashion.

- Operation of all storage units, regardless of their independence in satisfying processor requests, is synchronized, with no overlapping of read/write cycles. (This departure from the way modern storage units function is discussed in the paper.)
- Cycle duration is the same for all storage units.
- Input/output channels as well as central processing units are referred to as processors, although they can be distinguished by assigning special values to certain parameters (the tie-breaking probabilities).
- Each processor, i, can request use of a storage unit for only one cycle, and does so with probability p_i. Thus, the demand pattern of each processor is equivalent to a sequence of Bernoulli trials.
- If a processor fails to get use of a storage unit for a requested cycle, it automatically repeats its request for the next cycle. Thus, the sequences of Bernoulli trials are intermittently shifted forward, which activity can be regarded as a Markov chain.
- Only one request can be satisfied each cycle by one storage unit.

two processors and two storage units

Consider the case of two processors, A and B. Storage unit j ($j = 1, 2$) is requested for each cycle by processor A with probability p_{aj} and by processor B with probability p_{bj}. If both processors request the same storage unit, j, for the same cycle, processor A will win with probability Π_{aj} and processor B will win with probability Π_{bj}. Thus, $0 \leq p_{aj} + p_{bj} \leq 2$ and $\Pi_{aj} + \Pi_{bj} = 1$ for $j = 1, 2$. Each time a processor request is not satisfied, its refused request and all of its subsequent requests are postponed one cycle. Thus, we have two parallel sequences of Bernoulli trials, which are intermittently shifted forward.

The shifting process can be described as a finite Markov chain[1] with the five states shown in Table 1. Let P_{ij} ($i, j = 1, \cdots, 5$) be the conventional transition probability of going from state i to state j. The matrix P of transition probabilities P_{ij} is given by

$$P = \begin{bmatrix} 1 - p_{a1}p_{b1} - p_{a2}p_{b2} & p_{a1}p_{b1}\Pi_{b1} & p_{a1}p_{b1}\Pi_{a1} & p_{a2}p_{b2}\Pi_{b2} & p_{a2}p_{b2}\Pi_{a2} \\ 1 - p_{b1} & p_{b1}\Pi_{b1} & \boxed{p_{b1}\Pi_{a1}} & 0 & 0 \\ 1 - p_{a1} & p_{a1}\Pi_{b1} & p_{a1}\Pi_{a1} & 0 & 0 \\ 1 - p_{b2} & 0 & 0 & p_{b2}\Pi_{b2} & p_{b2}\Pi_{a2} \\ 1 - p_{a2} & 0 & 0 & p_{a2}\Pi_{b2} & p_{a2}\Pi_{a2} \end{bmatrix}$$

For example, to go from state 2 to state 3, which has probability P_{23} (shown boxed), we require that processor B demand a cycle on storage unit 1 (which has probability p_{b1}) and that processor A win the resulting conflict (which has probability Π_{a1}). We do not require a request by processor A, since state 2 implies this.

Table 1 Markov chain of five states

State	*Explanation*
1	Neither processor is delayed
2	Processor A is delayed on storage unit 1; B is using 1
3	Processor B is delayed on storage unit 1; A is using 1
4	Processor A is delayed on storage unit 2; B is using 2
5	Processor B is delayed on storage unit 2; A is using 2

The Markov chain represented by the matrix P is both irreducible and aperiodic.[2] Thus, if the matrix P is multiplied by itself many times, it converges to a matrix with identical rows:

$$\lim_{n\to\infty} P^n \equiv A = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 \\ P_1 & \cdot & \cdot & \cdot & P_5 \\ P_1 & \cdot & \cdot & \cdot & P_5 \\ P_1 & \cdot & \cdot & \cdot & P_5 \\ P_1 & \cdot & \cdot & \cdot & P_5 \end{bmatrix}$$

The elements P_j ($j = 1, \cdots, 5$) of the limit matrix A are the limiting probabilities that the system will be found in state j. By definition, the sum of the five probabilities $P_1, \cdots, P_5$ equals 1. Since the limiting operation converges to A, it follows that:

$$A \cdot P = A$$

This matrix operation represents five simultaneous equations with five unknowns, the limiting state probabilities, and can be readily solved by the techniques of matrix algebra. In general, there are n unknowns for n states.

the stretching factor computed

Processor A is delayed one cycle each time state 2 or 4 is entered. Therefore, in the limit, processor A is delayed $(P_2 + P_4)X$ cycles for every X cycles. Consequently, after X cycles, processor A has advanced only $X - (P_2 + P_4)X$ cycles, so that the stretching factor for processor A is $[1 - (P_2 + P_4)]^{-1}$. This factor can be interpreted as a ratio, T_a^*/T_a, where T_a is the time to do a certain task on processor A without contention and T_a^* is the corresponding time with contention. Similarly, for processor B, $T_b^*/T_b = [1 - (P_3 + P_5)]^{-1}$. By solving the transition matrix P for the limiting probabilities, P_i, we have:

$$\frac{T_a^*}{T_a} = \frac{(1 - S_1)(1 - S_2) + p_{a1}p_{b1}(1 - S_2) + p_{a2}p_{b2}(1 - S_1)}{(1 - S_1)(1 - S_2) + p_{a1}p_{b1}\Pi_{a1}(1 - S_2) + p_{a2}p_{b2}\Pi_{a2}(1 - S_1)} \quad (1)$$

and

$$\frac{T_b^*}{T_b} = \frac{(1 - S_1)(1 - S_2) + p_{a1}p_{b1}(1 - S_2) + p_{a2}p_{b2}(1 - S_1)}{(1 - S_1)(1 - S_2) + p_{a1}p_{b1}\Pi_{b1}(1 - S_2) + p_{a2}p_{b2}\Pi_{b2}(1 - S_1)} \tag{2}$$

where $S_1 = p_{a1}\Pi_{a1} + p_{b1}\Pi_{b1}$ and $S_2 = p_{a2}\Pi_{a2} + p_{b2}\Pi_{b2}$

two processors and N storage units

The foregoing analysis can be extended to cover the case of two processors and N storage units. The results are:

$$\frac{T_a^*}{T_a} = \frac{1 + \sum_{i=1}^{N} \frac{p_{ai}p_{bi}}{1 - S_i}}{1 + \sum_{i=1}^{N} \frac{p_{ai}p_{bi}\Pi_{ai}}{1 - S_1}} \tag{3}$$

$$\frac{T_b^*}{T_b} = \frac{1 + \sum_{i=1}^{N} \frac{p_{ai}p_{bi}}{1 - S_i}}{1 + \sum_{i=1}^{N} \frac{p_{ai}p_{bi}\Pi_{bi}}{1 - S_i}} \tag{4}$$

where

$$S_i = p_{ai}\Pi_{ai} + p_{bi}\Pi_{bi} \qquad i = 1, \cdots, N$$

p_{ai} = the probability that storage unit i is requested for any particular cycle by processor A, and p_{bi} = the similar figure for processor B.

Π_{ai} = the probability that A will be granted the storage unit if both A and B request storage unit i for the same cycle, and $\Pi_{bi} = 1 - \Pi_{ai}$.

one processor, one channel, and N storage units

A channel is distinguished from a processor by the value of the probability with which it prevails in obtaining use of a storage unit for a cycle, in the event that it and a processor both request the unit for the same cycle. Ordinarily, the channel has priority, so that this situation is equivalent to the case of two processors and N storage units if one processor is privileged over the other. Therefore, let A be the channel and B the processor; then $\Pi_{ai} = 1$ and $\Pi_{bi} = 0$ for all i. The stretching factors then become:

$$\frac{T_a^*}{T_a} = 1 \quad \text{(the channel is unaffected)} \tag{5}$$

$$\frac{T_b^*}{T_b} = 1 + \sum_{i=1}^{N} \frac{p_{ai}p_{bi}}{1 - p_{ai}} \tag{6}$$

three processors and one storage unit

The number of states increases from five to the seven shown in Table 2 as we add a third processor to the case of two processors and one storage unit; however, the number of independent parameters increases from three to eight. This means that explicit general formulas are more difficult to obtain and more cumbersome to use.

The fifteen system parameters are the following, of which only eight are independent:

- p_x is the probability that processor X requests a storage unit for any given cycle, where $X = a, b, c$. Define $q_x = 1 - p_x$.

Table 2 Markov chain of seven states

State	*Explanation*
1	No processor is delayed
2	Processor B is delayed; either A or C is using storage unit
3	Processor A is delayed; either B or C is using storage unit
4	Processor C is delayed; either A or B is using storage unit
5	Processors B and C are delayed; A is using storage unit
6	Processors A and C are delayed; B is using storage unit
7	Processors A and B are delayed; C is using storage unit

- Π_{xy} is the probability that if processors X and Y both request a storage unit for the same cycle, X prevails and Y is delayed, where $X, Y = a, b, c$.
- Π_{xyz} is the probability that if processors X, Y, and Z all request a storage unit for the same cycle, X prevails and Y and Z are delayed for $X, Y, Z = a, b, c$.

The forty-nine transition probabilities are specified in the Appendix.

For this and all other cases where at least three processors are involved, it is best to first substitute the numerical values of the parameters and then solve the associated set of linear simultaneous equations. However, explicit formulas are given for two particular situations.

In one case, one processor representing a channel, C, with an arbitrary storage demand rate and high priority, is involved with two processors, A and B, each having an arbitrary storage demand rate but lower priority than the channel. Thus, p_a, p_b, and p_c are arbitrary. A and B are given equal priority in a conflict between them by

$$\Pi_{ab} = \Pi_{ba} = \tfrac{1}{2}$$

Absolute priority is given to C in a conflict with either A or B by

$$\Pi_{ca} = \Pi_{cb} = 1$$

Absolute priority is given to C in a conflict with both A and B by

$$\Pi_{cab} = \Pi_{cba} = 1$$

Using these values, the following set of three simultaneous equations can be derived, from which the limiting state probabilities, P_2, P_3, and P_7, may be calculated:

$$\begin{aligned} &p_a q_b P_2 + (q_a p_b - 2q_c - 2p_a p_c)P_3 \\ &\qquad + (2q_c + p_a p_b + 2p_a p_c)(1 - P_7) = 2q_c \\ &p_a q_b p_c P_2 + q_a p_b p_c P_3 + (q_c + p_a p_b p_c)(1 - P_7) = q_c \qquad (7) \\ &(q_c + p_b p_c)P_2 - (q_c + p_a p_c)P_3 + p_c(p_a - p_b)(1 - P_7) = 0 \end{aligned}$$

Processor A is delayed whenever states 3, 6, and 7 are entered.

Similarly, processor B is delayed whenever states 2, 5, and 7 are entered. The values of P_4, P_5, and P_6 are zero, since the channel can never be delayed in contention with the other processors. P_1 can be obtained from the relation $P_1 + P_2 + P_3 + P_7 = 1$.

Thus,

$$T_a^*/T_a = [1 - (P_3 + P_6 + P_7)]^{-1}$$
$$T_b^*/T_b = [1 - (P_2 + P_5 + P_7)]^{-1} \tag{8}$$
$$T_c^*/T_c = 1$$

If the two processors, A and B, have the same storage demand rate, so that $p_a = p_b$, then the stretching factors for A and B are identical. For this case, the limiting state probabilities can be calculated directly from the following:

$$P_3 = \frac{\frac{1}{2}[p_a + 2q_a p_c]p_a q_c}{p_a[p_a + 2q_a p_c](q_c + p_a p_c) + q_a q_c(q_c + p_a^2 p_c)}$$
$$P_6 = 0$$
$$P_7 = \frac{p_a^2 p_c(1 + q_a p_c)}{p_a[p_a + 2q_a p_c](q_c + p_a p_c) + q_a q_c(q_c + p_a^2 p_c)} \tag{9}$$

In the second case, the three processors are identical in their storage demand rate; however, the priority scheme is arbitrary. Thus, $p_a = p_b = p_c = p$ and $q = 1 - p$.

Π_{ab}, Π_{ac}, Π_{bc}, Π_{abc}, Π_{bac}, and Π_{cab} are arbitrary. Processor A is delayed whenever states 3, 6, and 7 are entered. By symmetry, B and C are delayed by the same factor as A. Thus,

$$T_a^*/T_a = T_b^*/T_b = T_c^*/T_c = [1 - (P_3 + P_6 + P_7)]^{-1}$$

and the limiting probabilities are:

$$P_3 = \frac{\{(1 - pq\Pi_{ab})[p^2q^3(\Pi_{ba} + \Pi_{ca}) + p^3(1 + pq)(\Pi_{ba}\Pi_{cab} + \Pi_{ca}\Pi_{bac}) + p^3q(1 + 2q)\Pi_{ca}]}{[(1 - pq\Pi_{ba})(1 - pq\Pi_{ab}) + p^2q^2(\Pi_{ca} - \Pi_{ba})(\Pi_{ab} - \Pi_{cb})][q^3 + qp^2(1 + 2q) + p^3(1 + pq)]}$$
$$+ \frac{pq(\Pi_{ba} - \Pi_{ca})[p^2q^3(\Pi_{ab} + \Pi_{cb}) + p^3(1 + pq)(\Pi_{cb}\Pi_{abc} + \Pi_{ab}\Pi_{cab}) + p^3q(1 + 2q)\Pi_{cb}]\} \cdot q}{[(1 - pq\Pi_{ba})(1 - pq\Pi_{ab}) + p^2q^2(\Pi_{ca} - \Pi_{ba})(\Pi_{ab} - \Pi_{cb})][q^3 + qp^2(1 + 2q) + p^3(1 + pq)]}$$

$$P_6 = \frac{p^3(1 + pq)\Pi_{bac}}{q^3 + qp^2(1 + 2q) + p^3(1 + pq)}$$
$$P_7 = \frac{p^3(1 + pq)\Pi_{cab}}{q^3 + qp^2(1 + 2q) + p^3(1 + pq)} \tag{10}$$

the general case

The solution to the general case of M processors and N storage units is essentially the solution of a large number of linear equations. The number of states rises sharply as the number of processors increases. (There are 65 states and therefore 65 equations if $M = 4$ and $N = 2$.) However, it may be that several processors make equal demands upon the storage units, so that many of the limiting probabilities are identical. Thus, if all four processors are alike in demand rate and priority, only five different limiting

probabilities exist. Therefore, the analysis of a large number of systems may not be hopelessly complex if a few additional assumptions are made.

Numerical examples

two processors and two storage units

For the first of two examples, assume that processor A needs storage unit 1 for 40 percent of the time and storage unit 2 for 40 percent of the time. Processor B needs storage unit 1 for 10 percent of the time and storage unit 2 for 70 percent of the time. In addition, A is favored in the event of simultaneous requests for storage unit 1, because it wins conflicts with B four times out of five. In the same way, B is favored in requests for storage unit 2. Note that the utilization of storage unit 2 is not 110 percent, because the programs being executed by A and B have finite length. Furthermore, if A and B both wanted storage unit 2 all the time, the stretching factor would be 2 for each (assuming the contested cycles were assigned to A and B alternately). Thus we have

$$P_{a1} = 0.4 \qquad P_{a2} = 0.4$$
$$P_{b1} = 0.1 \qquad P_{b2} = 0.7$$
$$\Pi_{a1} = 0.8 \qquad \Pi_{a2} = 0.2$$
$$\Pi_{b1} = 0.2 \qquad \Pi_{b2} = 0.8$$

Working with Equations 1 and 2 we obtain

$$S_1 = (0.4)\ (0.8) + (0.1)\ (0.2) = 0.34$$
$$S_2 = (0.4)\ (0.2) + (0.7)\ (0.8) = 0.64$$

so that

$$\frac{T_a^*}{T_a} = \frac{(0.66)(0.36) + (0.4)(0.1)(0.36) + (0.4)(0.7)(0.66)}{(0.66)(0.36) + (0.4)(0.1)(0.8)(0.36) + (0.4)(0.7)(0.2)(0.66)}$$

$$= 1.53$$

$$\frac{T_b^*}{T_b} = \frac{(0.66\quad 0.36) + (0.4)(0.1)(0.36) + (0.4)(0.7)(0.66)}{(0.66)(0.36) + (0.4)(0.1)(0.2)(0.36) + (0.4)(0.7)(0.8)(0.66)}$$

$$= 1.12$$

Thus, a program taking one minute to be executed by processor A in the absence of B now takes 1.53 minutes. Note too that B is not slowed as much as A, because most of its activity is on storage unit 2, which favors B.

If we modify this problem so that A always wins use of a storage unit for a contested cycle, then

$$\Pi_{a1} = \Pi_{a2} = 1 \text{ and } \Pi_{b1} = \Pi_{b2} = 0$$

Thus

$$\frac{T_a^*}{T_a} = 1 \quad \text{and} \quad \frac{T_b^*}{T_b} = 1.84.$$

This serves as a "worst case" analysis for B.

two processors, one channel, one storage unit

For the second example, assume that processors A and B each need the storage unit 30 percent of the time, and the channel's demand is for 20 percent of the time. Thus,

$p_a = p_b = 0.3$ and $p_c = 0.2$

From Equation 9,

$P_3 = 0.115$
$P_6 = 0.$
$P_7 = 0.034$

Therefore, the stretching factors given by Equations 7 and 8 are

$$\frac{T_a^*}{T_a} = \frac{T_b^*}{T_b} = 1.17 \quad \text{and} \quad \frac{T_c^*}{T_c} = 1$$

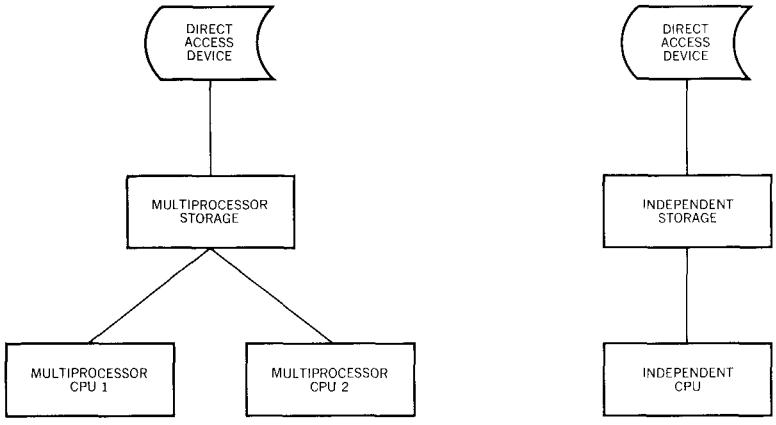
Simulation studies

The mathematical model is useful within the limitations of analytic techniques in general. The derivation of the analytic formulas is possible only within the framework of certain restrictive assumptions. Also, successful utilization of these formulas hinges upon a fairly precise knowledge of the various probabilities that combine to form the resultant equations. Therefore, a study was conducted to determine whether the mathematical model was valid in the general, nonrestrictive case or only within its rather limiting premises. The study was also intended to produce sets of probabilities required by the analytic formulas.

The mathematical model of the interacting processors and shared storage units is grounded on two main assumptions. First, the storage units operate cyclically and synchronously regardless of processor demand for access. This also implies that main storage has no potential for interleaving. Second, a processor's requests for access to storage are independent of prior demands (a processor's requests form a sequence of Bernoulli trials). Although critical for the derivation of the analytic formulas, the realism of the first of these assumptions is questionable when compared with the operation of an actual storage unit, which operates during a cycle only in the event of a processor request, which incorporates overlapping read/write cycles, and which may operate with a degree of interleaving. Likewise, the realism of the second assumption, and thus the entire model, may be brought into question when it is realized that processor requests for use of storage are not independent of one another.

Simulation of the multiprocessing environment was used to determine the predictive accuracy of the analytic technique. It should be emphasized that this work was not done to check the analytic formulas against a real system; it was important only to validate them against a system that was not based on the same restrictive assumptions, a system that, incidentally, incorporated all the relevant features and complexity of a real system. However, the multiprocessor model that was constructed

Figure 1 Multiprocesses model



using the General Purpose Simulation SYSTEM/360 does, in turn, involve certain assumptions of its own. The model consists of three central processing units, two that contend with each other for storage unit cycles and one that uses an independent storage unit. In addition, a contention and an independent storage unit are present together with a (removable) generalized direct-access device, as shown in Figure 1. Three major assumptions are embodied in the model:

- A teleprocessing-oriented instruction mix
- A fetch-restore storage cycle time of 1.5 microseconds
- When present, an input/output request rate of one storage request every 40 microseconds

The model operates at the instruction level, rather than simulating execution of complete job steps or programs, and it includes such processor features as eight-byte pre-fetching of instructions, and branching and accessing of data dependent upon the assigned instruction length. The storage units operate asynchronously and are structured to allow interleaving. The direct-access device, when operating, transmits at a rate of 200 kilobytes per second, generating the highest priority service requests in both the independent and the contention storage units at constant intervals of 40 microseconds. Each of the CPU's, in contrast, provides a new instruction immediately upon completion of use of the storage unit in processing the previous instruction. As these elements interact, storage contention occurs, and, in addition, the probabilities required for the analytic formulas previously discussed may be derived.

Table 3 presents the simulation results pertinent to verification of the analytic technique. The simulations included models both with and without the generalized direct-access device and storage units ranging in complexity from no interleaving to four-way interleaving. The accesses per instruction include both instruction

Table 3 Simulation results

			Accesses per instruction	*Instructions per second*		*Enhancement (percent)*	*Simulation stretching factor*	*Analytic stretching factor*
				CPU 1	*CPU 2*			
No interleaving	Independent processor	Without I/O	1.41	427,416			1	
		With I/O	1.42	409,357			1.044	1.043
	Multiprocessor	Without I/O	1.42	236,227	233,477	9.89	1.819	1.809
		With I/O	1.42	225,661	228,282	10.89	1.883	1.879
2-way interleaving	Independent processor	Without I/O	1.42	488,219			1	
		With I/O	1.41	477,920			1.0215	1.0225
	Multiprocessor	Without I/O	1.42	329,746	333,363	35.82	1.473	1.580
		With I/O	1.41	327,176	329,577	37.41	1.487	1.623
4-way interleaving	Independent processor	Without I/O	1.42	519,313			1	
		With I/O	1.41	510,050			1.012	1.016
	Multiprocessor	Without I/O	1.42	388,213	387,406	49.35	1.339	1.436
		With I/O	1.41	388,619	383,771	51.43	1.344	1.467

and data accesses. Although the independent system executed a greater number of instructions than either CPU of the multiprocessor, the instruction rate of the multiprocessor was generally greater than that of the independent system, due to the considerable contention for storage unit cycles exhibited by the multiprocessing CPU's. The enhancement percentages provide a measure of this increased rate with respect to the independent system. These figures are arrived at by subtracting the independent processor instruction rate from the total multiprocessor instruction rate and dividing the result by the independent instruction rate.

It seems logical to begin a verification of the analytic technique with inspection of the simple example in which a processor contends only with a channel for their shared storage unit. This corresponds to the case of one processor, one channel, and N stores, considered previously. With only one storage unit, N is equal to 1, and Equation 6 reduces to

$$\frac{T_b^*}{T_b} = 1 + \frac{p_a p_b}{1 - p_a} \tag{11}$$

The probability, p_b, of a storage access by the processor may be determined by first calculating the maximum number of requests for service that a single processor may make in one second. With two-way interleaving, it may be assumed that one-half of the processor's requests spend the minimum amount of time, 0.75 microsecond, in use of storage. The remaining requests cannot effectively utilize the interleaving potential and are forced to spend 1.5 microseconds while being serviced by main storage.

$$\text{Maximum accesses/second} = \frac{1 \text{ access}}{(0.5)(0.75)+(0.5)(1.5) \text{ microseconds}}$$

$$= 888{,}889 \text{ accesses/second}$$

The actual instruction rate of the independent processor without input/output interference is shown in Table 3 to be 488,219 instructions per second. This figure varies according to the instruction mix.

Table 3 also shows that approximately 1.42 accesses to main storage were required for each instruction. This figure seems reasonable if it is realized, first, that the four-byte instructions, which constitute the majority of the instruction set of the simulation model, each require one data access, and, second, that two such instructions can be fetched per access, given a storage width of eight bytes. The deviation of this figure from the expected value of 1.5 depends on the relative percentage of two-byte instructions, which do not access storage for data, and of six-byte instructions, which require two such data accesses. This value of accesses per instruction varies according to different equipment configurations and instruction mixes.

The probability of a storage access by the processor is, then, equal to

$$p_b = \frac{(1.42 \text{ accesses/instruction})\ (488{,}219 \text{ instructions/second})}{888{,}889 \text{ potential accesses/second}}$$

$$= 0.780$$

Note that for a storage with no interleaving, $p_b = 0.904$; for a storage with four-way interleaving, $p_b = 0.691$.

Since the channel demands service of the storage once every 40 microseconds, there are 25,000 actual channel requests per second. Each of these requests requires only one storage access,

so that p_a, the probability of a storage access by the channel, may be expressed as

$$p_a = \frac{(1.00 \text{ accesses/request})\ (25{,}000 \text{ requests/second})}{888{,}889 \text{ potential accesses/second}} = 0.028$$

Substitution into Equation 11 yields

$$\frac{T_b^*}{T_b} = 1 + \frac{(0.028)(0.780)}{1 - 0.028} = 1.0225$$

The corresponding simulation stretching factor may be calculated by division of the independent processor's instruction throughput without I/O interference by the same processor's throughput with such interference. This fraction may be seen, from Table 3, to be

$$\frac{488{,}219 \text{ instructions/second}}{477{,}920 \text{ instructions/second}} = 1.0215$$

In the case of one processor, one channel, and one store, then, it appears that the analytic and simulation approaches yield nearly identical results.

The verification of a slightly more complex analytic formula presents itself in the case of two processors and N stores, where N is again made equal to 1. In reality, this configuration would be equivalent to a multiprocessing system with no I/O devices. If it is assumed that each processor has an equal probability of requesting this storage unit for any particular cycle and, further, that a cycle under contention is granted with equal probability to one of the two processors, then Equations 3 and 4 become equal, and p_a assumes equality with p_b. Letting $p_a = p_b = p$,

$$\frac{T_a^*}{T_a} = \frac{T_b^*}{T_b} = \frac{1 - p + p^2}{1 - p + (\frac{1}{2})p^2}$$

Substitution of the previously determined probability, $p_b = 0.780$, yields

$$\frac{T_a^*}{T_a} = \frac{T_b^*}{T_b} = 1.580$$

The corresponding simulation stretching factor may be produced by dividing the independent processor throughput by the average throughput of a single CPU in the multiprocessor.

$$\text{stretching factor (simulation)} = \frac{488{,}219 \text{ instructions/second}}{331{,}554 \text{ instructions/second}}$$

$$= 1.473$$

This indicates that the analytic formula yields a value about seven percent higher than simulation. This error was fairly consistent over several runs with differing parameters, indicating that this

difference arises from the assumptions made and not from statistical variation of the simulation. Considering the grossness of the assumptions, it is a very modest error.

The final verification deals with the case of three processors and one storage unit, where one of the three processors assumes the attributes of a channel. Obeying the premises of the previous example, evaluation of the analytic stretching factor merely involves substitution of the processor and the channel probabilities, 0.780 and 0.028, respectively, into Equations 8 and 9.

$$\frac{T_a^*}{T_a} = \frac{T_b^*}{T_b} = \frac{1}{1 - (0.363 + 0 + 0.021)} = 1.623$$

Simulation of the same case produces a stretching factor through division of the independent processor throughput with I/O by the average throughput of a multiprocessor CPU without I/O. That is,

$$\frac{488{,}219 \text{ instructions/second}}{328{,}376 \text{ instructions/second}} = 1.487$$

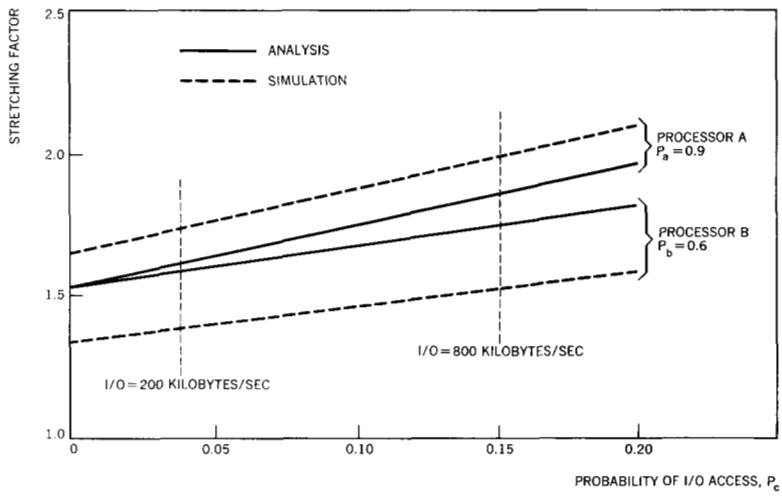
In this example, then, the results of the analytic technique are approximately nine percent above the simulation figures.

Although the calculations in the three foregoing examples involve only the simulation data for a memory unit with two-way interleaving, Table 3 also displays the analytic and simulation values for the environments in which the storage allows either no interleaving or four-way interleaving. The conclusion to be drawn from the correspondence of the three sets of figures is that it is reasonable to expect that an analytic approach will yield results within ten percent of those derived from an actual simulation of the same problem. In this particular instance, it should be noted that the percentage deviation of the analytic from the simulation figures increases as the degree of storage interleaving increases. This merely highlights the fact that the assumptions of the analytic technique become increasingly inaccurate as the complexity of the model is augmented.

If the premises of the second example are relaxed to the extent that the two contending processors request storage cycles with differing probabilities, calculation of the analytic stretching factor for the case of three processors and one storage unit involves the resolution of the three simultaneous equations of Equation 7.

Figure 2 shows the results for the case of unequal probabilities, $p_a = 0.9$ and $p_b = 0.6$. No storage interleaving is assumed. The stretching factor is plotted as a function of increasing channel activity. It is important to note that the simulation model indicates a difference between the stretching factors for processors A and B under the condition of no I/O requests, which the analytic model does not predict. With increasing I/O activity, a difference in the stretching factors for the two processors appears in the analytic model, but the spread remains more modest than the simulation results. However, the average stretching factors from the two methods are nearly the same.

Figure 2 Stretching factor for two processors with contending I/O activity



Summary

The analytic approach appears to be useful in providing approximate stretching factors for storage contention. However, if the desired results must be much more accurate than 10 to 15 percent, it is usually necessary to resort to simulation; the advantages gained through the speed of the analytic technique ordinarily are balanced by its inability to mirror changes in model complexity as readily as simulation.

CITED REFERENCES

1. W. Feller, *An Introduction to Probability Theory and Its Applications*, Chapter XV, Second Edition, John Wiley & Sons, New York, New York (1957).
2. F. R. Gantmacher, *The Theory of Matrices*, Volume 2, 87–98, Chelsea Publications Company, New York, New York (1959).

Appendix

For the case of three processors and one storage unit, the identities among the tie-breaking probabilities are as follows:

$$\Pi_{ab} = 1 - \Pi_{ba},\ \Pi_{ac} = 1 - \Pi_{ca},\ \Pi_{bc} = 1 - \Pi_{cb}$$
$$\Pi_{abc} = \Pi_{acb},\ \Pi_{bac} = \Pi_{bca},\ \Pi_{cab} = \Pi_{cba}$$
$$\Pi_{abc} + \Pi_{bac} + \Pi_{cab} = 1.$$

The forty-nine transition probabilities P_{ij} found in the 7-state transition matrix P are:

$$P_{11} = 1 - p_a p_b q_c - p_a q_b p_c - q_a p_b p_c - p_a p_b p_c$$
$$P_{12} = p_a p_b q_c \Pi_{ab} + q_a p_b p_c \Pi_{cb}$$
$$P_{13} = p_a p_b q_c \Pi_{ba} + p_a q_b p_c \Pi_{ca}$$
$$P_{14} = p_a q_b p_c \Pi_{ac} + q_a p_b p_c \Pi_{bc}$$
$$P_{15} = p_a p_b p_c \Pi_{abc}$$
$$P_{16} = p_a p_b p_c \Pi_{bac}$$
$$P_{17} = p_a p_b p_c \Pi_{cab}$$
$$P_{21} = q_a q_c$$
$$P_{22} = p_a q_c \Pi_{ab} + q_a p_c \Pi_{cb}$$
$$P_{23} = p_a q_c \Pi_{ba}$$
$$P_{24} = q_a p_c \Pi_{bc}$$
$$P_{25} = p_a p_c \Pi_{abc}$$
$$P_{26} = p_a p_c \Pi_{bac}$$
$$P_{27} = p_a p_c \Pi_{cab}$$
$$P_{31} = q_b q_c$$
$$P_{32} = p_b q_c \Pi_{ab}$$
$$P_{33} = p_b q_c \Pi_{ba} + q_b p_c \Pi_{ca}$$
$$P_{34} = q_b p_c \Pi_{ac}$$
$$P_{35} = p_b p_c \Pi_{abc}$$
$$P_{36} = p_b p_c \Pi_{bac}$$
$$P_{37} = p_b p_c \Pi_{cab}$$
$$P_{41} = q_a q_b$$
$$P_{42} = q_a p_b \Pi_{cb}$$
$$P_{43} = p_a q_b \Pi_{ca}$$
$$P_{44} = p_a q_b \Pi_{ac} + q_a p_b \Pi_{bc}$$
$$P_{45} = p_a p_b \Pi_{abc}$$
$$P_{46} = p_a p_b \Pi_{bac}$$
$$P_{47} = p_a p_b \Pi_{cab}$$
$$P_{51} = 0$$
$$P_{52} = q_a \Pi_{cb}$$
$$P_{53} = 0$$
$$P_{54} = q_a \Pi_{bc}$$
$$P_{55} = p_a \Pi_{abc}$$
$$P_{56} = p_a \Pi_{bac}$$
$$P_{57} = p_a \Pi_{cab}$$
$$P_{61} = 0$$
$$P_{62} = 0$$
$$P_{63} = q_b \Pi_{ca}$$
$$P_{64} = q_b \Pi_{ac}$$
$$P_{65} = p_b \Pi_{abc}$$
$$P_{66} = p_b \Pi_{bac}$$
$$P_{67} = p_b \Pi_{cab}$$
$$P_{71} = 0$$
$$P_{72} = q_c \Pi_{ab}$$
$$P_{73} = q_c \Pi_{ba}$$
$$P_{74} = 0$$
$$P_{75} = p_c \Pi_{abc}$$
$$P_{76} = p_c \Pi_{bac}$$
$$P_{77} = p_c \Pi_{cab}$$