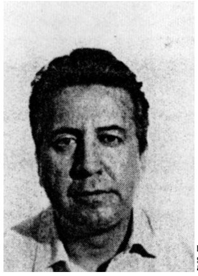
Key words: Multiprocessor systems, nonexponential service times, performance modeling and analysis, processing efficiency, simulation models.

Simulation experiments for performance analysis of multiple-bus multiprocessor systems with nonexponential service times



Ibrahim H. Onyuksel
Department of Computer Science
George Mason University
Fairfax, VA 22030.

K. B. Irani
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109.

IBRAHIM H. ONYUKSEL received Yuksek Muhendis (Dipl.-Ing.) degree in electrical engineering from Istanbul Technical University, Istanbul, Turkey, in 1969, the M.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1977, and the Ph.D. degree in computer, information and control engineering from the University of Michigan, Ann Arbor, MI in 1985. From 1971 to 1972 he was with the Turkish State Radio and Television Network, and from 1972 to 1973 he was with Marmara Scientific and Industrial Research Institute in Turkey, where he worked on the RF communication hardware. Currently he is an Assistant Professor in the Department of Computer Science at Northern Illinois University, DeKalb, IL. Before joining NIU, he was on the faculty of Northwestern University, Evanston, IL and George Mason University, Fairfax, VA. His current research interests include performance evaluation of computer and communication systems, and queueing network models.

Dr. Onyuksel is a member of the IEEE and ACM. He was on the program committee of COMPSAC 1986 and 1987.

KEKI B. IRANI received the B.S. degree in mechanical engineering and the B.S. degree in electrical engineering from the University of Bombay, Bombay, India, in 1946 and 1947, respectively, and the M.S.E. degree in electrical engineering in 1949 and the Ph.D. degree in 1953, both from the University of Michigan, Ann Arbor, MI.

From 1950 to 1957 he was a Research and Design Engineer for Philips Telecommunication Industries, The Netherlands, where he worked on coaxial telephone systems. He joined the faculty of the Department of Electrical Engineering, University of Kansas in 1956, and the University of Michigan faculty in 1961. He is now with the University of Michigan, where he is a Professor in the Department of Electrical Engineering and Computer Science. He has worked in the areas of electrical power systems, electrical networks, communications, control and computers. His current research interests are computer architecture, distributed computer systems, databases and artificial intelligence.

Dr. Irani is a member of the IEEE, ACM, Sigma Xi, and Tau Beta.

ABSTRACT

A simulation model (program) is constructed for performance analysis of multiple-bus multiprocessor systems with shared memories. It is assumed that the service time of the common memory is either hypo- or hyperexponentially distributed. Processing efficiency is used as the performance index. To investigate the effects of different service time distributions on the system performance, comparative results are obtained for a large set of input parameters. The simulation results show that the error in approximating the memory access time by an exponentially distributed random variable is less than 6% if the coefficient of variation is less than 1, but it increases drastically with this factor if it is greater than 1.

INTRODUCTION

In early multiprocessor systems, a crossbar switch was used to connect the processors to the common memory. For example, a widely known crossbar system is C.mmp multi-minicomputer [1]. The performance of a crossbar multiprocessor system has been analyzed in recent years [2]-[6]. However, crossbar interconnection networks are becoming less and less interesting due to their complexity and their cost. Recent proposals and implementations show that a more attractive alternative would be a bus-structured interconnection network [7], [8].

The performance of bus-oriented multiprocessor systems has been studied by many researchers. Fung and Torng [9] developed a deterministic model for the analysis of memory contention and bus conflicts in multiple-bus multiprocessor systems. Goyal and Agerwala [10] proposed two generic classes of multiple-bus systems, and they analyzed the performance of these systems using the independence approximation introduced by Hoogendoorn [4].

In a way similar to that suggested by Enslow [11], a multiple-bus multiprocessor system, as depicted in Figure 1, can be described by its characteristics as follows:

- The multiprocessor system contains two or more processors of comparable capabilities. Each processor has its own local memory unit.
- All processors share access to a common memory, which consists of several modules.
- Processors and common memory modules are connected by multiple buses.
- The allocation of common resources to processors is controlled by a controller unit.

Processors execute segments of programs in their local memories until they need to access the common memory. When a processor requests access the common memory, it computes an address, including a memory module number, and then signals the controller for a connection to the referenced module. Requests for connection are assumed to be independent from one processor to another, and more than one processor may request access simultaneously. A processor can access the common memory via one of the time-shared buses if the referenced module is free and a bus is available for connection. The *configuration* of this type of systems is usually denoted by a triple $p \times m \times b$, where p, m, and b are number of processors, number of memory modules, and number of buses, respectively.

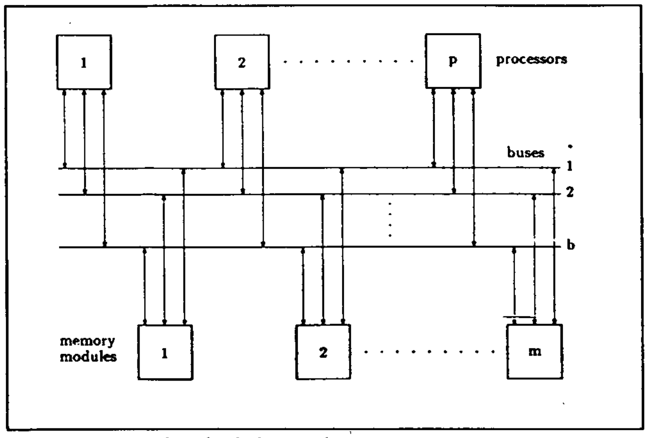


Figure 1. A typical multiple-bus multiprocessor system.

The main reason for using multiprocessor systems and dividing the common memory into several modules is to achieve better performance for the proposed system. In theory, a multiprocessor system with N independent processors can compute a given problem at most N times faster than any one of the processors can. However, theoretical speedup cannot be achieved in practice because of sharing the common resources among the processors.

Section 2 presents a closed queueing network model for the performance analysis of a multiple-bus multiprocessor system. Section 3 shows a simple way to use hypo- and hyperexponential distributions to match the first two moments of a random variable. Finally, in Section 4, our simulation model is explained and the effects of the input parameters on the multiprocessor system's performance are discussed.

Performance Modeling

To analyze the performance of a multiprocessor system with N independent processors under conflicts, the behavior of the system can be modeled by a closed queueing network with N classes of customers, two stages of parallel servers (processors and common memory modules), and several passive resources (buses for processor-memory connection).

The queueing network model of the multiple-bus multiprocessor system (Fig. 1) can be described by the following assumptions:

(1) When a processor requests access the common memory, a connection is immediately established between the processor and the referenced module, provided that the referenced module is not being accessed by another processor and a bus is available for connection.

(2) A processor cannot have another memory requested if its present request has not been granted.

(3) The duration between the completion of a request and the generation of the next request to the common memory is an independent, exponentially distributed random variable with the same mean value of $1/\lambda$ for all processors.

(4) The duration of an access by a processor to the common memory is an independent, identically distributed random variable with the same mean value of $1/\mu$ for all memory modules.

(5) The probability of a request for access from a processor to a common memory module is independent of the module and it is equal to $1/m$.

If a queueing network model satisfies assumption (5), then it is called a *uniform reference model* (URM). Although this assumption considerably simplifies the analysis, it may not be very realistic for some systems. Several researchers have attempted to solve the problem with nonuniform access probabilities. However, their methods are applicable only to small-scale systems [12]–[15].

The goal of the analysis of the queueing network model is to determine the values of a performance measure for a given set of input parameters. A performance measure is merely an index which can be used to represent the performance of a system. In this paper, *processing efficiency* (PE), which is equal to the expected value of the percentage of ACTIVE processors, is used as a direct measure for the "computing power" of a multiprocessor system—A processor is called active if it is executing instructions in its private memory, and an active processor is neither accessing nor waiting to get access to the common memory. Most of the other performance measures for the queueing network model of a multiple-bus multiprocessor system are related to PE with very simple equations [16], and an exact, closed-form solution for PE with exponentially distributed memory access times is obtained by Irani and Onyuksel [17].

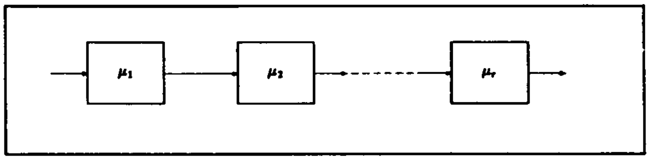


Figure 2: The hypoexponential server.

Non-exponential Service Times

In this paper, we assume that the service time of the common memory is either hypo- or hyperexponentially distributed, because these distributions are sufficient to match the first two moments of a given random variable. Let X be a random variable. If X has an exponential distribution, then *the coefficient of variation* for X is known to be $C_X = \sigma_X/E[X] = 1$, where $E[X]$ is the expected value and σ_X is the standard deviation of X. If $C_X \neq 1$ then it is possible to find a hypoexponential distribution for $C_X < 1$ and a hyperexponential distribution for $C_X > 1$, which exactly matches the first two moments of X.

An r-stage Erlang distribution can be realized by an r-stage serial server such that each stage of the server has exponentially distributed service time with the same service rate for all stages. Erlang distribution can be generalized by relaxing the restriction that each stage of the server has the same service rate. This is called the hypoexponential distribution. Figure 2 illustrates the r-stage hypoexponential server, where each stage of the server has exponentially distributed service time with the mean value of $1/\mu_i$ for the ith stage. Within the service facility, at most one of the stages may be occupies by a customer and no new customer may enter the server until the previous one departs. Customers enter from the the left and depart to the right.

Let $\mu_i = k_i\mu$ for $i = 1,\ldots,r$ be the exponential service rate of the ith stage of an r-stage hypoexponential server, and let Y be the probability distribution to this server with $E[Y] = 1/\mu$. It is known that

$$E[Y] = \sum_{i=1}^{r} \frac{1}{\mu_i} = \frac{1}{\mu} \sum_{i=1}^{r} \chi_i \qquad (1)$$

$$\sigma_Y^2 = \sum_{i=1}^{r} \frac{1}{\mu_i^2} = \frac{1}{\mu^2} \sum_{i=1}^{r} \chi_i^2 \qquad (2)$$

where $\chi_i = 1/k_i$ for $i = 1,\ldots,r$. Equation (1) yields

$$\sum_{i=1}^{r} \chi_i = 1 \qquad (3)$$

where $\chi_i \leqslant (k_i \geqslant 1)$ for $i = 1,\ldots,r$. By definition of C_Y and by equations (1) and (2), it is obtained

$$C_Y^2 = \sum_{i=1}^{r} \chi_i^2. \qquad (4)$$

It can be easily shown that C_Y reaches its minimum value for $\chi_1 = \cdots = \chi_r = 1/r$ (Erlang distribution), which yields $C_Y = 1/\sqrt{r}$. In other words, if X is a probability distribution with $C_X \in [1/\ r, 1]$, then there exists an r-stage hypoexponential distribution Y which exactly matches the first two moments of X. Combining equations (3) and (4) yields a more suitable expression for C_Y as follows:

$$C_Y^2 = 1 - 2 \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \chi_i \chi_j \qquad (5)$$

An r-stage hyperexponential distribution can be realized by an r-stage parallel server such that each stage of the server has exponentially distributed service time. Figure 3 illustrates the r-stage hyperexponential server. As for the hypoexponential server, at most of the stages may be occupied by a customer and upon entry into the service facility the customer proceeds to service stage i with probability α_i.

Let $\mu_i = k_i\mu$ for $i = 1,\ldots,r$ be the exponential service rate of the ith stage of an r-stage hyperexponential server, and let Z be the probability distribution corresponding to this server with $E[Z] = 1/\mu$. It is known that

$$E[Z] = \sum_{i=1}^{r} \frac{\alpha_i}{\mu_i} = \frac{1}{\mu} \sum_{i=1}^{r} \alpha_i \chi_i \qquad (6)$$

$$\sigma_Z^2 = 2\left(\sum_{i=1}^{r} \frac{\alpha_i}{\mu_i^2}\right) - (E[Z])^2$$

$$= \frac{2}{\mu^2}\left(\sum_{i=1}^{r} \alpha_i \chi_i^2\right) - \frac{1}{\mu^2} \qquad (7)$$

where $\chi_i = 1/k_i$ for $i = 1,\ldots,r$. Equation (6) yields

$$\sum_{i=1}^{r} \alpha_i \chi^i = 1 \qquad (8)$$

where $\chi_i \leqslant 1/\alpha_i (k_i \geqslant \alpha_i)$ for $i = 1,\ldots,r$. By definition of C_Z and by equations (6) and (7), it is obtained

$$C_Z^2 = 2\left(\sum_{i=1}^{r} \alpha_i \chi_i^2\right) - 1 \qquad (9)$$

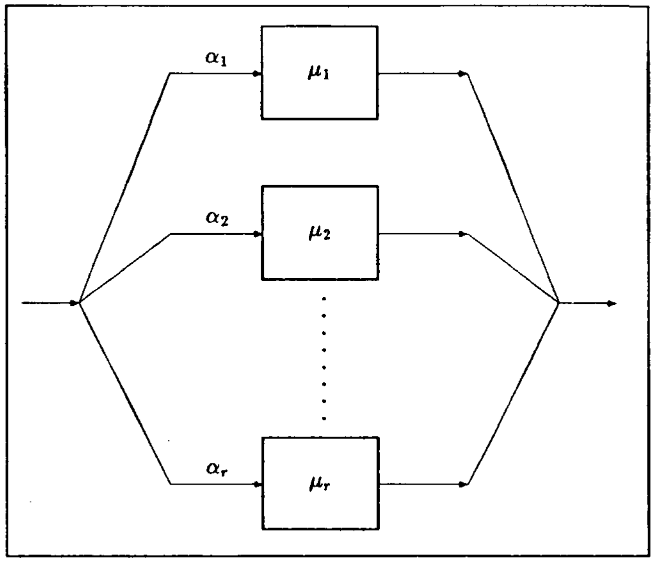


Figure 3. The hyperexponential server.

A Special Case

If $\alpha_1 = \cdots = \alpha_r = 1/r$ then equations (8) and (9) can be simplified as follows:

$$\sum_{i=1}^{r} \chi_i = r,$$

$$C_Z^2 = \frac{2}{r}\left(\sum_{i=1}^{r} \chi_i^2\right) - 1.$$

$$= (2r-1) - \frac{4}{r} \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \chi_i \chi_j.$$

The preceding expressions show that C_Z reaches its maximum value for $\chi_1 = r$ and $\chi_i = 0$ for $i = 2,\ldots,r$, which yields $C_z = \sqrt{2r-1}$. In other words, if X is an arbitrary probability distribution with $C_z \in [1, \sqrt{2r-1}]$, then there exists an r-stage hyperexponential distribution Z (with equal branching probabilities) which exactly matches the first two moments of X.

Simulation Experiments

Because of its generality and its simplicity, the most popular approach to analyze the performance of a computer system is to use a simulation technique. However, to run a simulation program is usually very costly, and its execution time is goverend by the number of sample points which directly determines the error on the final output statistics. In most cases, the error can be reduced by additional computational time.

For the performance analysis of a multiple-bus multiprocessor system with a common memory, a simulation program was constructed. The program was run on the Michigan Terminal System (MTS) for several multiprocessor configurations with hypo- or hyperexponential memory access times.

For random number generation, the pseudo-random sequences generated by a subroutine (on MTS) were used. This is called a *multiplicative congruential generator* which generates a sequence $\{u_i\}$ uniformly distributed on the interval (0,1). To eliminate the dependency between various events of the simulation experiment, different *seed* numbers (randomly selected) are assigned to each independent sequence of events. For example, the interval between subsequent access requests of a processor to the common memory is independent of the others, so that different seed numbers are used for each processor.

For the simulation program, there are two types of probability distributions to generate, namely, the uniform distribution and the exponential distribution. Since hypo- and hyperexponential distributions can be represented by a serial or a parallel combination of several exponential distributions, they do not need to be generated separately.

Let U and X be uniformly distributed random variables in the intervals (0,1) and (a,b), respectively, and let Y be an exponentially distributed random variable with parameter λ. It is then shown in Reference [18] that

$$X = a + (b - a)\, U, \tag{10}$$

$$Y = -\frac{1}{\lambda}(\ln U). \tag{11}$$

If the modules of the common memory are numbered from 1 to m and if I is a random integer uniformly distributed between 1 to m, then by assigning $a = 1$ and $b = m + 1$ in equation (10), I can be generated from X as follows:

$$X = 1 + mU \longrightarrow I = 1 + [mU].$$

Let V be an r-stage hypoexponential distribution with parameters $\mu_1, \ldots, \mu_r$. If $\{Y_i\}$ is a sequence of independent, exponentially distributed random variables corresponding to the stages of the hypoexponential distribution such that $V = \sum_{i=1}^{r} Y_i$, and if $\{U_i\}$ is a sequence of independent, uniformly distributed random variables in the interval (0,1), then equation (11) yields

$$V = -\sum_{i=1}^{r} \frac{1}{\mu_i}(\ln U_i).$$

If V is an r-stage Erlang distribution with parameter μ, then $\mu_1 = \cdots = \mu_r = r\mu$. Thus, the preceeding equation yields

$$V = -\frac{1}{r\mu}\ln\left(\prod_{i=1}^{r} U_i\right).$$

Let W be an r-stage hyperexponential distribution with parameters $\mu_1, \ldots, \mu_r$. If a stage in the parallel server is chosen uniformly, and if I is a random integer uniformly distributed between 1 to r, then

$$W = Y_I = -\frac{1}{\mu_I}(\ln U_2) \quad \text{with} \quad I = 1 + [rU_1],$$

where U_1 and U_2 are independent, uniformly distributed random variables in the interval (0,1) and μ_I is chosen uniformly from $\{\mu_1, \ldots, \mu_r\}$ by the index variable I.

The objective of our simulation experiments is to investigate the behavior of the multiple-bus multiprocessor system on its equilibrium condition. Since the behavior of the simulation model does not represent the transient behavior of the system, the data observed during this period are discarded. Let N_s be the number of sample points used to compute the final statistics and N_t be the discarded data points. Thus, the program runs until $N = N_t + N_s$ data points are observed. In order to estimate the transient period, we made a number of preliminary pilot runs (with different seed numbers for each run) and compared the final statistics at various "ages." These simulation experiments showed that if we discard the initial $N_t = 1{,}000$ sample data points of each run, the effect of transient period and the selection of seed numbers on the final statistics becomes less than 1%. Of course, the selection of N_t seems rather arbitrary, but if N is sufficiently large, then it is reasonable to believe that the error, which is made by considering the system in equilibrium after N_t data points, is negligible.

It is clear that the simulation error on the final statistics decreases as N_s increases. However, if N_s is too large, then running the simulation program will become very costly. Therefore, a lower bound for N_s must be estimated for a given error percentage. Suppose $E[Y]$ is chosen as a performance index. If the samples for $E[Y]$, $Y_1, \ldots, Y_{N_s}$, are statistically independent, then for sufficiently large values of N_s, it is shown in Reference [18] that

$$\Pr\left[\bar{Y} - \frac{z_{\alpha/2}s_Y}{\sqrt{N_s}} \leqslant E[Y] \leqslant \bar{Y} + \frac{z_{\alpha/2}s_Y}{\sqrt{N_s}}\right] = 1 - \alpha, \tag{12}$$

where

$\bar{Y} = \frac{1}{N_s}\sum_{i=1}^{N_s} Y_i$ is the sample mean, $s_Y^2 = \frac{1}{N_s - 1}\sum_{i=1}^{N_s}(Y_i - \bar{Y})^2$

is the sample variance, and $z_{\alpha/2}$ is the upper $100(\alpha/2)$ percentile of the standard normal distribution.

The interval $K_{1-\alpha} = (\bar{Y} - z_{\alpha/2}s_Y N_s^{-1/2}, \bar{Y} + z_{\alpha/2}s_Y N_s^{-1/2})$ is said to be $100(1-\alpha)$ percentage confidence interval for $E[Y]$. Equation (12) shows that $E[Y]$ is contained in the interval with probability $(1-\alpha)$. For the confidence interval considered above, the length L of the interval may be written as, $L = 2z_{\alpha/2}s_Y N_s^{-1/2}$, and it follows

$$N_s = \left(\frac{2z_{\alpha/2}s_Y}{L}\right)^2. \tag{13}$$

For given α and s_Y values, N_s can be determined by the preceeding equation so that the confidence interval will have a prescribed length. In general, s_Y is not known in advance, but it can be estimated by a pilot run.

For the simulation experiments, $\alpha = 0.05$ and $(L/2\bar{Y})$ were chosen. First, a pilot run was made by using $N_s = 100{,}000$ ($N_t = 10{,}000$) sample points for the $2 \times 2 \times 2$ system with $\rho = \lambda/\mu = 1$ and $C_s = 1/\sqrt{2}$, which is the coefficent of variation for the service time S of the common memory: PE = 45.37% and $\sigma_{PE} = 35.80\%$ were obtained. This yields $L = 0.91\%$ for a 1% simulation error with 95% confidence, and by equation (13), we obtain $N_s \cong 24{,}000$. However, $N_s = 50{,}000$ with $N_t = 5{,}000$ were selected to obtain more precise results.

Tables 1 and 2 give the simulation statistics (mean value, standard deviation, and 95% confidence interval) on PE for a family of bus-sufficient (BS) systems with p,m,b = 2,4,6,8, ρ = 1, and C_S = 0.90, $1/\sqrt{2}$. Simulation results are also compared with the exponential service times (C_S = 1)[1]. Let us define $\Delta PE = (PE - PE1)/PE1$ with PE1 be the PE of a system with an exponential server. In fact, ΔPE is a direct measure of the effect of C_S on the system performance. Since $1/\sqrt{2} \leq C_S < 1$ for both cases of the example, the service time of the common memory can be approximated by a two-stage hypoexponential distribution with $\mu_1 = k_1\mu$ and $\mu_2 = k_2\mu$: For C_S = 0.90, k_1 = 1.12 and k_2 = 9.41, and for $C_S = 1/\sqrt{2}$, $k_1 = k_2 = 2$.

The values of ΔPE in Tables 1 and 2 show that, for $1/\sqrt{2} \leq C_S < 1$, the error in approximating the service time of the common memory by an exponential distribution is less than 5%. Indeed, simulation results show that the maximum error is about 4.51 ± 0.43% with 95% confidence (the 8 × 6 × 6 system with P = 1).

At the extreme point, C_S = 0 (deterministic access times). For the 6 × 4 × 2 bus-deficient (BD) system, the exact results for C_S = 1 and the simulation results for C_S = 0[2] are compared for ρ = 0.1,0.5,1.0. This comparison yielded ΔPE = 0.64, 5.55, 4.78%, respectively. These values show that even for the extreme case, the maximum error is less than 6%.

To run a simulation program for large-scale multiprocessor systems with less than 1% error is computationally inhibitive. Even for small-scale systems our simulation program took, on the average, 100 seconds of CPU time (on MTS) to produce one data point for N = 55,000. This seems to be a drawback in using a simulation technique for the performance analysis.

To investigate the characteristics of PE for $C_S > 1$, the simulation program was run for C_S = 1.10 and $\sqrt{2}$. The results are tabulated in Tables 3 and 4. Since $1 < C_S < \sqrt{3}$ for both cases of the example, the service time of the common memory can be approximated by a two-stage hyperexponential distribution with a uniform selection for each stage, where $\mu_1 = k_1\mu$ and $\mu_2 = k_2\mu$: For C_S = 1.10, k_1 = 0.76 and k_2 = 1.48, and for $C_S = \sqrt{2}$, k_1 = 0.59 and k_2 = 3.41.

Table 1. Simulation statistics on PE for ρ = 1 and C_S = 0.90.

p	m	b	PE	σ_{PE}	K_{95}	PE1	ΔPE(%)
2	2	2	44.88	36.49	(44.56,45.20)	44.44	0.98± 0.72
4	2	2	34.65	25.65	(34.42,34.87)	34.33	0.94± 0.66
6	2	2	27.11	19.53	(26.94,27.28)	26.70	1.54± 0.64
8	2	2	21.64	15.47	(21.51,21.78)	21.43	0.98± 0.63
2	4	2	47.40	35.90	(47.09,47.72)	47.06	0.72± 0.67
4	4	4	41.70	26.00	(41.47,41.93)	41.34	0.87± 0.56
6	4	4	36.61	21.04	(36.42,36.79)	36.14	1.29± 0.51
8	4	4	32.17	17.79	(32.01,32.33)	31.66	1.60± 0.51
2	6	2	48.22	35.86	(47.91,48.54)	48.00	0.46± 0.66
4	6	4	44.32	25.86	(44.09,44.55)	44.05	0.60± 0.52
6	6	6	40.73	21.11	(40.54,40.91)	40.29	1.10± 0.46
8	6	6	37.50	18.32	(37.34,37.66)	36.80	1.90± 0.43
2	8	2	48.60	35.73	(48.29,48.92)	48.48	0.24± 0.65
4	8	4	45.77	25.80	(45.55,46.00)	45.48	0.64± 0.49
6	8	6	43.04	21.34	(42.85,43.23)	42.55	1.14± 0.45
8	8	8	40.54	18.40	(40.38,40.70)	39.76	1.96± 0.40

[1] *The exact results for the exponential case are obtained from Reference [19].*
[2] *Simulation results for fixed access times are obtained from Reference [16].*

Table 2. Simulation statistics on PE for ρ = 1 and $C_S = 1/\sqrt{2}$.

p	m	b	PE	σ_{PE}	K_{95}	PE1	ΔPE(%)
2	2	2	45.39	35.80	(45.08,45.71)	44.44	2.13± 0.71
4	2	2	35.52	24.83	(35.30,35.74)	34.33	3.47± 0.64
6	2	2	27.75	18.77	(27.59,27.92)	26.70	3.94± 0.62
8	2	2	21.95	14.69	(21.82,22.08)	21.43	2.43± 0.61
2	4	2	47.70	35.85	(47.38,48.01)	47.06	1.36± 0.67
4	4	4	42.46	25.45	(42.24,42.68)	41.34	2.70± 0.53
6	4	4	37.70	20.52	(37.52,37.88)	36.14	4.31± 0.50
8	4	4	33.05	17.19	(32.90,33.20)	31.66	4.38± 0.47
2	6	2	48.52	35.50	(48.21,48.83)	48.00	1.08± 0.65
4	6	4	45.00	25.37	(44.78,45.23)	44.05	2.15± 0.51
6	6	6	41.81	20.73	(41.63,42.00)	40.29	3.78± 0.46
8	6	6	38.46	17.73	(38.30,38.62)	36.80	4.51± 0.43
2	8	2	48.90	35.48	(48.59,49.21)	48.48	0.86± 0.64
4	8	4	46.19	25.33	(45.97,46.41)	45.48	1.57± 0.48
6	8	6	43.86	20.74	(43.68,44.04)	42.55	3.07± 0.42
8	8	8	41.45	17.96	(41.29,41.61)	39.76	4.25± 0.40

Table 3. Simulation statistics on PE for ρ = 1 and C_S = 1.10.

p	m	b	PE	σ_{PE}	K_{95}	PE1	ΔPE(%)
2	2	2	44.12	37.13	(43.80,44.45)	44.44	-0.73± 0.73
4	2	2	33.95	26.54	(33.72,34.19)	34.33	-1.10± 0.68
6	2	2	26.25	20.29	(26.07,26.43)	26.70	-1.68± 0.67
8	2	2	21.10	16.29	(20.96,21.24)	21.43	-1.54± 0.65
2	4	2	46.96	36.56	(46.64,47.28)	47.06	-0.21± 0.68
4	4	4	40.86	26.65	(40.62,41.09)	41.34	-1.17± 0.57
6	4	4	35.31	21.78	(35.12,35.50)	36.14	-2.31± 0.53
8	4	4	31.06	18.35	(30.89,31.22)	31.66	-1.90± 0.52
2	6	2	47.75	36.14	(47.43,48.07)	48.00	-0.52± 0.67
4	6	4	43.40	26.32	(43.17,43.63)	44.05	-1.48± 0.52
6	6	6	39.62	21.65	(39.43,39.81)	40.29	-1.66± 0.47
8	6	6	35.84	18.78	(35.68,36.01)	36.80	-2.61± 0.45
2	8	2	48.38	35.88	(48.06,48.69)	48.48	-0.22± 0.65
4	8	4	44.96	26.06	(44.74,45.19)	45.48	-1.14± 0.49
6	8	6	41.97	21.59	(41.78,42.16)	42.55	-1.37± 0.45
8	8	8	39.16	19.05	(39.00,39.33)	39.76	-1.51± 0.41

Table 4. Simulation statistics on PE for ρ = 1 and $C_S = \sqrt{2}$.

p	m	b	PE	σ_{PE}	K_{95}	PE1	ΔPE(%)
2	2	2	43.15	38.39	(42.82,43.49)	44.44	-2.91± 0.75
4	2	2	32.38	27.91	(32.14,32.62)	34.33	-5.67± 0.70
6	2	2	25.13	21.87	(24.94,25.32)	26.70	-5.88± 0.71
8	2	2	20.23	17.60	(20.07,20.38)	21.43	-5.60± 0.72
2	4	2	46.27	37.14	(45.94,46.59)	47.06	-1.68± 0.69
4	4	4	39.12	27.82	(38.87,39.36)	41.34	-5.37± 0.59
6	4	4	33.50	22.64	(33.31,33.70)	36.14	-7.32± 0.54
8	4	4	29.24	19.41	(29.07,29.41)	31.66	-7.65± 0.54
2	6	2	47.16	36.73	(46.84,47.48)	48.00	-1.75± 0.67
4	6	4	41.95	27.30	(41.71,42.19)	44.05	-4.78± 0.54
6	6	6	37.76	22.72	(37.56,37.96)	40.29	-6.27± 0.50
8	6	6	34.13	19.81	(33.95,34.30)	36.80	-7.25± 0.48
2	8	2	47.98	36.49	(47.66,48.30)	48.48	-1.04± 0.66
4	8	4	43.73	26.86	(43.49,43.97)	45.48	-3.84± 0.53
6	8	6	40.32	22.49	(40.12,40.52)	42.55	-5.25± 0.47
8	8	8	37.29	19.59	(37.12,37.46)	39.76	-6.21± 0.43

The values of ΔPE in Tables 3 and 4 show that for $1 < C_S \leq \sqrt{2}$, the error in approximating the service time of the common memory is less than 8 %. The values in Tables 1-4 yield the following observation:

If $C_S < 1$ then PE>PE1, and if $C_S > 1$ then PE<PE1. Thus, to increase the PE of a system, σ_S need to be decreased for the service time of the common memory.

CONCLUSION

In this paper, the performance analysis of a typical multiple-bus multiprocessor system is extended beyond the exponential distribution and a simulation model is developed for hypo- and hyperexponentially distributed memory access times.

Processing efficiency is used as a primary performance measure. For a large set of values, the effect of C_S on PE investigated and the comparative results are presented. If the coefficient of variation for the service time of the common memory of the multiple-bus multiprocessor system is less than 1, then our results show that approximating the service time by an exponential distribution will not produce a large percentage of error on the system performance. Even in the worst case (constant service time), ΔPE is less than 6% with 95% confidence.

REFERENCES

[1] W.A. Wulf and C.G. Bell, "C.mmp—A multi-miniprocessor," in Proc. *AFIPS*, 1972, pp. 765-777.

[2] D.P. Bhandarkar, "Analysis of memory interference in multiprocessors," *IEEE Trans. Comput.*, vol. C-24, pp. 897-908, Sept. 1975.

[3] F.S. Baskett and A.J. Smith, "Interference in multiprocessor computer systems with interleaved memory," *Commun. ACM*, vol. 19, pp. 327-334, Jun. 1976.

[4] C.H. Hoogendoorn, "A general model for memory interference in multiprocessors," *IEEE Trans. Comput.*, vol. C-26, pp. 998-1005, Oct. 1977.

[5] B.R. Rau, "Interleaved memory bandwidth in a model of a multiprocessor computer system," *IEEE Trans. Comput.*, vol. C-28, pp. 678-681, Sept. 1979.

[6] D.W. Yen, J.H. Patel, and E.S. Davidson, "Memory interference in synchronous multiprocessor systems," *IEEE Trans. Comput.*, vol. C-31, pp. 1116-1121, Nov. 1982.

[7] R.J. Swan et al., "CM*—A modular multi-microprocessor," in *Proc. AFIPS*, 1977, pp. 637-644.

[8] J.V. Levy, "Buses: The skeleton of computer structures," in *Computer Engineering: A DEC View of Hardware System Design*, C.G. Bell, J.C. Mudge, and J.E. McNamara, eds., Digital Equipment Corp., 1978.

[9] F. Fung and H.C. Torng, "On the analysis of memory conflicts and bus contentions in a multiple-microprocessor system," *IEEE Trans. Comput.*, vol. C-28, pp. 28-37, Jan. 1979.

[10] A. Goyal and T. Agerwala, "Performance analysis of future shared storage systems," *IBM J. Res. Develop.*, vol. 28, pp. 95-108, Jan. 1984.

[11] P.H. Enslow, "Multiprocessor organization—A survey," *ACM Comput. Surveys*, vol.9, pp. 103-129, Mar. 1977.

[12] A.S. Sethi and N. Deo, "Interference in multiprocessor systems with localized memory access probabilities," *IEEE Trans. Comput.*, vol. C-28, pp. 157-163, Feb. 1979.

[13] K.O. Siomalas and B.A. Bowen, "Performance of crossbar multiprocessor systems," *IEEE Trans. Comput.*, vol. C-32, pp. 689-695, Jul. 1983.

[14] H.C. Du and J.L. Baer, "On the performance of interleaved memories with nonuniform access probabilities," in *Proc. 1983 Int'l Conf. on Parallel Processing*, pp. 429-436.

[15] D. Towsley, "An approximate analysis of multiprocessor systems," in *Proc. 1983 ACM Sigmetrics Conf. on Meas. and Mod. Comput. Syst.*, pp. 207-213.

[16] M. Ajmone Marsan and M. Gerla, "Markov models for multiple-bus multiprocessor systems," *IEEE Trans. Comput.*, vol. C-31, pp. 239-248, Mar. 1982.

[17] K.B. Irani and I.H. Onyuksel, "A closed-form solution for the performance analysis of multiple-bus multiprocessor systems," *IEEE Trans. Comput.*, vol. C-33, pp. 1004-1012, Nov. 1984.

[18] H. Kobayashi, *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*, Addison-Wesley, 1978.

[19] I.H. Onyuksel, Markovian *Queueing Network Models for Performance Evaluation of Multiple-Bus Multiprocessor Systems*, Ph.D. Dissertation, The Univ. of Michigan, Ann Arbor, Mich., May 1985.