Modeling of Buffered Multiple Bus Multiprocessor Systems in a Non-Uniform Traffic Environment

M. Atiquzzaman
Dept. of Computer Science
La Trobe University
Melbourne 3083, Australia.
atiq@LATCS1.lat.oz.au

M.A. Sayeed
Dept. of Electrical Engg.
King Fahd Univ. of Pet. & Minerals
Dhahran 31261, Saudi Arabia.

Abstract

Performance modeling of multiple-bus systems is usually carried out under the assumption of uniform memory reference model. Hot spots arising in multiprocessor systems give rise to non-uniform memory references. It is known that a hot spot memory reference pattern results in a significant degradation in the performance of a buffered multistage interconnection network. The aim of this research is to study the effect of hot spots on the performance of buffered multiple-bus systems. Analytical models based on Markov chains have been developed to determine the bandwidth of buffered multiple-bus systems in a hot spot memory reference pattern. The models assume that unsuccessful memory requests are queued in the buffers at the memory modules. Furthermore, processors with outstanding requests are not allowed to generate new requests. The model allows a fast and inexpensive method of performance evaluation of a multiple bus system in the case of a hot spot memory reference pattern.

1 Introduction

A multiprocessor system essentially consists of a number of processors (or processors and memories) interconnected using an interconnection network. Commonly used networks include the multiple-bus, multistage, hypercube, tree, etc. Multiple-bus systems have been found to be suitable for systems having a small number of processors. Multiple bus systems are modular and has the advantages of easy expansion and high fault tolerance.

Performance of multiple-bus systems is limited due to bus and memory contentions. Different conflict resolution strategies are used to resolve such conflicts [1]. Commonly used criteria for performance measurement of multiple-bus systems include bandwidth, probability of request acceptance, and processor utilization.

Multiple bus systems can be either buffered or unbuffered. In an unbuffered system, unsuccessful processor requests (due to contention) during a cycle are discarded. The processors have to resubmit the requests at subsequent cycles. In a buffered system, the unsuccessful requests are queued in buffers at the memory modules, and are resubmitted to the memory modules in the next cycle. Performance evaluation of both unbuffered [2, 3, 4, 5, 6, 7, 8] and buffered [9, 10, 11, 12] crossbar and multiple bus systems have been extensively studied. A good review of multiple bus systems appears in [13]. Most authors assume a uniform memory request pattern, whereby a memory request generated by a processor is equally likely to be directed to any of the memory modules [2, 3, 14, 15, 16, 17, 18, 19].

The uniform memory reference pattern is rather restrictive in real-world situations. Studies have shown that non-uniform request patterns may arise in multiprocessor systems. Favorite request pattern in multiple bus systems has been analyzed in [4], while consecutive request pattern has been studied in [20]. Successive requests with local referencing has been discussed in [9, 21]. Hot spot request pattern in multistage interconnection networks have been studied in [22, 23, 24, 25, 26, 27, 28], Other non-uniform request patterns can be found in [8, 29].

One of the reasons for hot request memory reference pattern is the use of shared variables used for locking, synchronization, pointers to shared queues, etc. These are indivisible primitives and must be stored in a single shared memory, thereby giving rise to a hot spot in the system. Combining has been shown to alleviate the problem in multistage interconnection networks when the hot spot is due to a single memory location. It is, however, not applicable if a memory module is itself a hot module. Therefore, it is important to study the performance of a multiple bus system in the presence of a hot memory reference pattern.

Performance evaluation of unbuffered multiple-bus and crossbar systems under hot spot request pattern have been reported in [30] and [31] respectively. *The objective of this paper is to develop analytical models to study the performance of buffered multiple-bus systems in the presence of a hot spot memory reference pattern.* Unsuccessful requests, in such a system, are buffered in first-in-first-out queues at the memory modules. Moreover, processors with outstanding requests are blocked from issuing further requests. The discrete-time Markov chain has been used for the modeling, and simulations have been carried out for the cases where the state space of the chain becomes too large to be handled with reasonable effort.

The rest of the paper is organized as follows. The modeling assumptions regarding the operation of the system are described in Section 2, followed by the Markov chain performance models in Section 3. Performance results obtained from models and simulations are presented in Section 4, and concluding remarks appear in Section 5.

0-7803-1814-5/94 $4.00 © 1994 IEEE

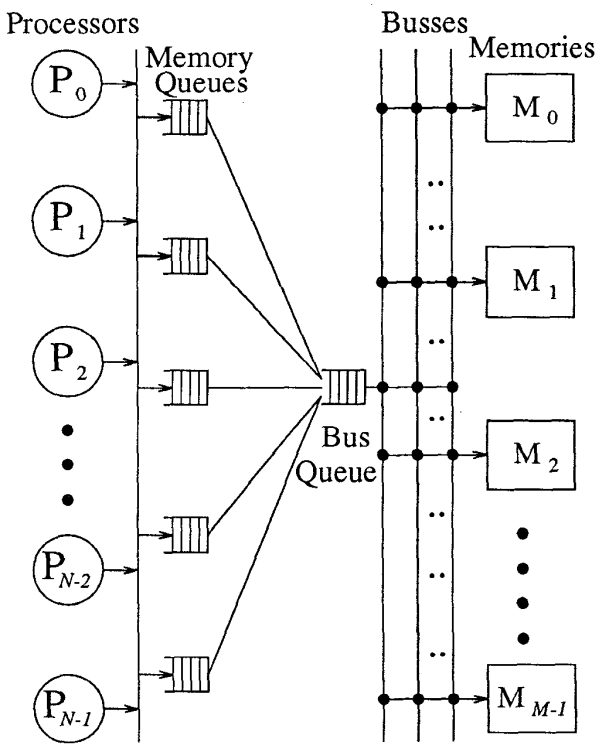

Figure 1: A queued multiple bus system

2 Modeling Assumptions

To keep the Markov chain model simple and tractable, several simplifying assumptions regarding the operation of the system are made. The following are the assumptions that will be used in the model described in Section 3.

- The system (Figure 1) consists of a multiple bus system with N processors $(\mathrm{P}_0, \mathrm{P}_1, \ldots, \mathrm{P}_{N-1})$ and M memory modules $(\mathrm{M}_0, \mathrm{M}_1, \ldots, \mathrm{M}_{M-1})$ and B busses. Without loss of generality, we assume M_0 to be the hot memory module.

- We assume *synchronous* operation of the system, i.e., the generation of requests by the processors and the servicing of the requests by the memory modules occur at clock cycles. The clock cycle (τ) is split into two phases τ_1 and τ_2 corresponding to the generation and servicing of requests. Processors generate requests at the beginning of τ_1, and the memory modules complete the servicing of the requests at the end of τ_2.

- Requests are assumed to be *spatially independent*, i.e., requests generated by a processor are independent of requests generated by other processors.

- A processor having an outstanding request is *blocked*, and can not generate a new request until the previous one has been served. A processor with no outstanding request generates a new request at the beginning of τ_1 with probability r. Because of requests being queued, the average request rate of a processor is, in effect, less than r.

- The probability that the generated request is for the hot memory module (HM) or a non-hot memory module (NHM) are h and $\bar{h} = (1-h)/(M-1)$ respectively.

- Processor requests generated during τ_1 of a cycle are put in the *memory queues* corresponding to the requested modules. During τ_2, a memory module services one outstanding (if any) request from its buffer.

- In the case of bus conflicts, processors requests to be serviced are chosen at random from the outstanding requests. B requests are presented to B different memory modules.

3 Performance Evaluation

Markov chain modeling of an $N \times M$ system even in a uniform memory reference pattern results in a large number of states which is not analytically tractable [9, 11]. In a hot spot memory reference pattern, the number of states is even larger. To illustrate the performance modeling of a multiple bus system in a hot spot memory reference pattern, we develop Markov chain models for systems where the number of processors or memories is small [9]. Due to the large number of states in the chain, we rely on simulation results for systems having a large number of processors and memories.

3.1 Modeling a $2 \times M$ system

In this section, we develop a Markov chain model for the average memory bandwidth of a system having two processors, M memory modules and, two busses. We assume that each memory module has a buffer of size N. The state of the system is defined by an M-tuple $(S_0, S_1, S_2, \ldots, S_{M-1})$, where $S_i, 0 \leq i \leq M-1$, is the number of outstanding memory requests for M_i at the end of τ_1. Note that $\sum_{i=0}^{M-1} S_i \leq 2$. Since we are interested only in the bandwidth of the system, we define the states of the system by seven 2-tuple equivalent states (see Figure 2), $\pi_{2,M} = \{\pi_0, \pi_1, \pi_2, \ldots, \pi_6\}$, based on the number of outstanding requests for the HM and NHMs $\mathrm{M}_j, 1 \leq j \leq M-1$ as follows.

π_0 = (2,0): two outstanding requests for the HM
π_1 = (1,0): one outstanding request for the HM
π_2 = (1,1): one outstanding request for the HM and one for the NHM
π_3 = (0,0): No outstanding requests
π_4 = (0,1,1): Two outstanding requests for two different NHMs
π_5 = (0,1): One outstanding request for a NHM
π_6 = (0,2): Two outstanding requests for the same NHM

Note that an *equivalent* state is composed of several states of the system. For example, the equivalent state π_6 is composed of the following states.

$$(0,2,0,\ldots,0),(0,0,2,0,\ldots,0),$$
$$(0,0,0,2,0,\ldots,0),\ldots(0,0,\ldots,0,2)$$

Having defined the states of the system, the transition probabilities between the states will be determined next. The transition probabilities will be represented by a 7×7 matrix, $P_{2,M} = \{p_{i,j}, 0 \leq i,j \leq 6\}$. The state transition from state i to state j will be represented by $p_{i,j}$. The Markov

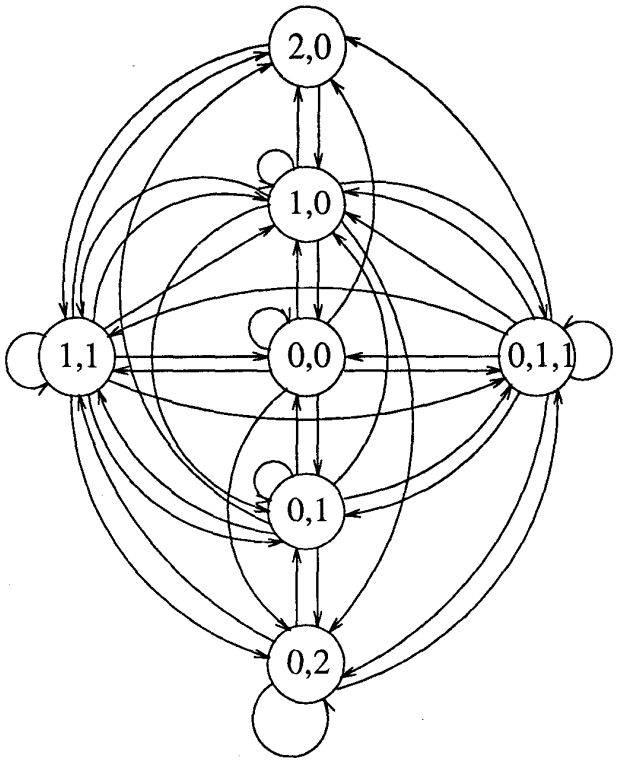

Figure 2: Markov diagram of a 2 processors, M memories system for $r < 1$

chain with all the possible transitions are given in Figure 2. Each arc represents a transition from one state to another state. When the system is in state π_0, a hot memory request is serviced. In the next cycle, the processor whose request was served in the previous cycle places a request to the HM or a NHM with probabilities rh and $r(1-h)$ respectively. The probability that it does not request is $(1-h)$. Therefore, the transition probabilities are given by $p_{0,0} = rh$, $p_{0,2} = r(1-h)$, and $p_{0,1} = (1-h)$. Otehr transition probabilities can be derived similarly. The transition probability matrix $P_{2,M}$ is given by

$$P_{2,M} = \begin{bmatrix} rh & (1-r) & \dots & 0 \\ r^2h^2 & 2r(1-r)h & \dots & r^2\bar{h}^{\,2}(M-1) \\ r^2h^2 & 2r(1-r)h & \dots & r^2\bar{h}^{\,2}(M-1) \\ r^2h^2 & 2r(1-r)h & \dots & r^2\bar{h}^{\,2}(M-1) \\ r^2h^2 & 2r(1-r)h & \dots & r^2\bar{h}^{\,2}(M-1) \\ r^2h^2 & 2r(1-r)h & \dots & r^2\bar{h}^{\,2}(M-1) \\ 0 & 0 & \dots & r\bar{h} \end{bmatrix}$$

It is possible to make a transition form any state back to the same state in a finite number of transitions and hence the chain in aperiodic. The chain is also irreducible. The chain is thus ergodic and hence possesses a unique stationary probability distribution of the different states. We solve the following two equations

$$\pi_{2,M} = \pi_{2,M} P_{2,M} \tag{1}$$

$$\sum_{i=0}^{6} \pi_i = 1 \tag{2}$$

Figure 3: Markov diagram of an N processors, 2 memories system

to obtain the stationary probability distributions $\pi_{2,M} = \{\pi_0, \pi_1, \dots, \pi_6\}$. For example,

$$\begin{aligned} \pi_1 &= [r^2h^2(1-r\bar{h}\,)]/\beta \\ \pi_2 &= [rh(1-r)(2-rh)(1-r\bar{h}\,)]/\beta \\ &\vdots \quad \vdots \quad \vdots \\ \pi_5 &= [r^2\bar{h}^{\,2}(M-1)(M-2)(1-rh)]/\beta \\ \pi_6 &= [r\bar{h}\,(1-r)(M-1)(2-r\bar{h}\,)(1-rh)]/\beta \end{aligned}$$

where, $\beta = r^2h^2(1-r\bar{h}\,)+(1-rh)(1-r\bar{h}\,)+r^2\bar{h}^{\,2}(M-1)(1-rh)$. The number of memory modules that can be accessed concurrently in state $\pi_i, 0 \le i \le 6$, will be represented by μ_i. For example, $\mu_1 = 1$ and $\mu_4 = 2$. The average memory bandwidth of the buffered $2 \times M$ system having two busses is therefore, given by

$$AMBW(2, M, 2) = \sum_{i=0}^{6} \mu_i \pi_i \tag{3}$$

where, $AMBW(N, M, B)$ is the average memory bandwidth of an $N \times M$ system having B busses.

3.2 Modeling an $N \times 2$ system

In this section, we develop a model to determine the average memory bandwidth of a system having N processors, two memories and two busses for $r = 1$. Since there are only two memories, the state of the system can be described by a 2-tuple (S_0, S_1), where S_0 and S_1 are the number of requests for the HM and the NHMs respectively. For an $N \times 2$ system, the total number of states is equal to all the possible partitions of N into two groups. Since there can be $N+1$ such partitions, the total number of possible states is $N+1$. For example, a 4×2 system has the states $\pi_{4,2} = \{(4,0), (3,1), (2,2), (1,3),$ and $(0,4)\}$. For a 9×2 system, the states are $\pi_{9,2} = \{(9,0), (8,1), (7,2), (6,3), (5,4), (4,5), (3,6), (2,7), (1,8), (0,9)\}$. The states of an $N \times 2$ system are $\pi_{N,2} = \{(N,0), (N-1,1), (N-2,2), \dots, (1,N-1), (0,N)\}$ as shown in the Markov chain diagram in Figure 3. When the system is in state $(N,0)$, a hot memory request is serviced and it goes to the intermediate state $(N-1,0)$ The next request is to the HM or the NHM with probabilities h and $\bar{h} = 1-h$ respectively. Thus the next state of the system is $(N,0)$ or $(N-1,1)$ with probabilities $p_{0,0} = h$ and $p_{0,1} = \bar{h}$ respectively. Other transition probabilities can be derived similarly. Two memory requests are serviced per cycle in all the states except $(N,0)$ and $(0,N)$ when one request is serviced. Therefore, $\mu_0 = \mu_N = 1$ and $\mu_i = 2, 1 \le i \le N-1$. The transition probability matrix for Figure 3 is given by

$$P_{N,2} = \begin{bmatrix} h & 1-h & \dots & 0 & 0 \\ h^2 & 2h(1-h) & \dots & 0 & 0 \\ 0 & h^2 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 2h(1-h) & (1-h)^2 \\ 0 & 0 & \dots & h & 1-h \end{bmatrix}$$

The Markov chain is ergodic and thus a stationary probability distribution $\pi_{N,2} = \{\pi_0, \pi_1, \ldots, \pi_N\}$ exists. Solving the set of equations (4) and (5)

$$\pi_{N,2} = \pi_{N,2} P_{N,2} \tag{4}$$

$$\sum_{i=0}^{N} \pi_i = 1 \tag{5}$$

we get

$$\begin{aligned} \pi_0 &= \frac{\alpha - 1}{\alpha^{2N} - 1} \\ \pi_1 &= \alpha(\alpha + 1)\frac{\alpha - 1}{\alpha^{2N} - 1} \\ &\vdots \quad \vdots \quad \vdots \\ \pi_i &= \alpha^2 \pi_{i-1}, \quad 2 \leq i \leq N - 1 \end{aligned}$$

where, $\alpha = (1 - h)/h$. The bandwidth is then found by substituting the values of μs and πs in

$$AMBW(N, 2, 2) = \sum_{i=0}^{N} \mu_i \pi_i \tag{6}$$

to obtain

$$AMBW(N, 2, 2) = \begin{cases} 1 + \frac{\alpha^{2N-1} - \alpha}{\alpha^{2N} - 1} & \text{for } \alpha \neq 1 \\ 2 - \frac{1}{N} & \text{for } \alpha = 1 \end{cases} \tag{7}$$

The AMBW for $r = 1$ is obtained as described above. The Markov chain for $r < 1$ is too complex because of the large number of possible states. As an example, the chain for a 3×2 system is shown in Figure 4. Consequently, we obtain the results for such a case using simulation.

3.3 Modeling an $N \times M$ System

Consider first a 4×4 system under the uniform memory reference pattern and $r = 1$. The number of states in such a system is equal to the number of equivalence classes in a decreasing list partition of four requests into four groups, where a group can be empty. The total number of partitions for the above case is five.

For an $N \times M$ system under the uniform memory reference pattern, the total number of partitions of N into M parts is given [32] by $\sum_{n=1}^{M} P(N, n)$, where $P(N, n)$ is given by the recurrence relation

$$P(N, n) = \sum_{i=1}^{n} P(N - n, i) \tag{8}$$

where, $P(N, 1) = P(N, N) = 1$. The N requests from the N processors can go to any number of memory modules out of the M modules. If we denote a state in a decreasing list format, then such a list $(i_1, i_2, \ldots, i_M)$, $i_1 \geq i_2 \geq i_3 \ldots \geq i_M$, where the entries sum up to N, describes a unique partition of N. Each of the unique partitions will be called an equivalent state. There are 15 equivalent states for a 7×7 system in the presence of a uniform memory reference pattern.

For a hot spot reference pattern, we redefine the representation of the states. A state is now represented by $(k, i_1, i_2, \ldots, i_{M-1})$, where $i_1, i_2, \ldots, i_{M-1}$ are arranged in a decreasing list. $k, 0 \leq k \leq N$, denotes the number of requests for the HM and $i_1, i_2, \ldots, i_{M-1}$ are the number of requests for the NHMs. The $N - k$ requests are to be partitioned into $M - 1$ NHMs. Therefore, the total number of states (Q) in such a representation is

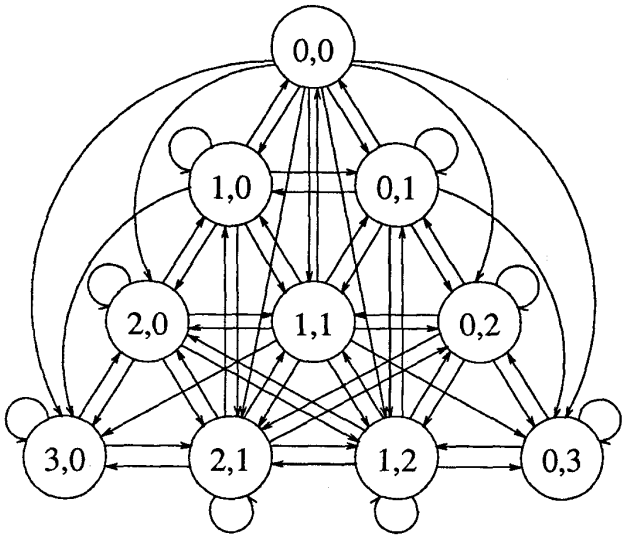

Figure 4: Markov diagram of a 3 processors, 2 memories system for $r < 1$

$$Q = 1 + \sum_{k=0}^{N-1} \sum_{n=1}^{\min(M-1, N-k)} P(N - k, n) \tag{9}$$

The number of transitions from a state of the Markov chain varies from one to a maximum of Q. The transition matrix is of size $Q \times Q$. It is tedious to solve $(Q+1)$ equations to determine the stationary probability distribution. Simulation was therefore, used to determine the bandwidth of $N \times M$ systems where N and M are greater than two.

4 Results

Results obtained for $2 \times M$, $N \times 2$, and $N \times M$ multiple bus systems using the models described in the previous sections are presented here. Figure 5 shows the variation in the average memory bandwidth of a 2×10 system having two busses, as a function of the hot spot probability, for different request rates. For this system, $h = 0.1$ corresponds to the uniform memory reference case. For high processor request rates ($0.6 \leq r \leq 1.0$), an increase in the hot spot probability results in contention at the hot memory module, resulting in a sharp decrease in the bandwidth. For low processor request rates ($0.1 \leq r \leq 0.5$) the bandwidth does not degrade significantly with an increase in the hot spot probability. This is due to little contention for the HM at low values of processor request rates. It should be noted that processors with blocked requests can not generate new requests, resulting in a drop in the effective processor request rate. The effective request rate decreases with an increase in the blocking at the memory modules. This decrease in the effective request rate with an increase in the blocking is analogous to a feedback approach to prevent the buffers from overflowing.

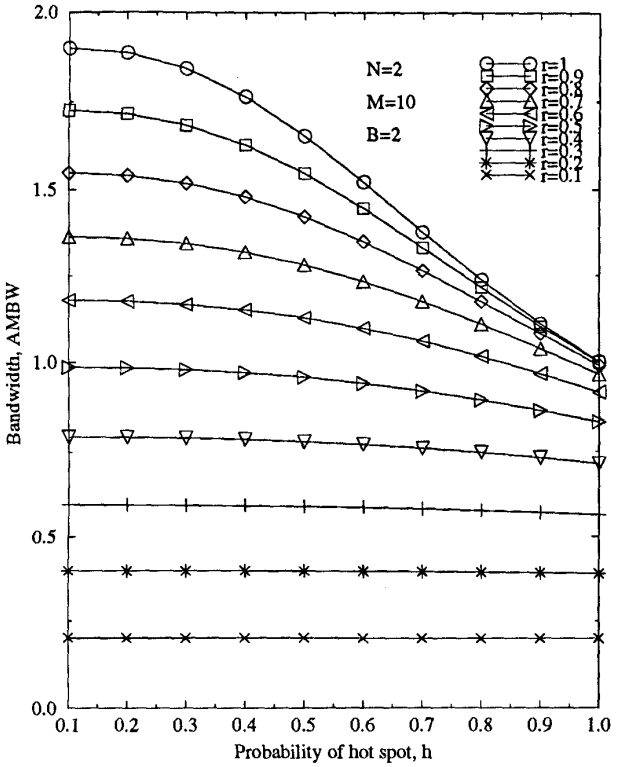

Figure 5: Bandwidth vs. hot spot probability for 2 processors M memories system for $r < 1$

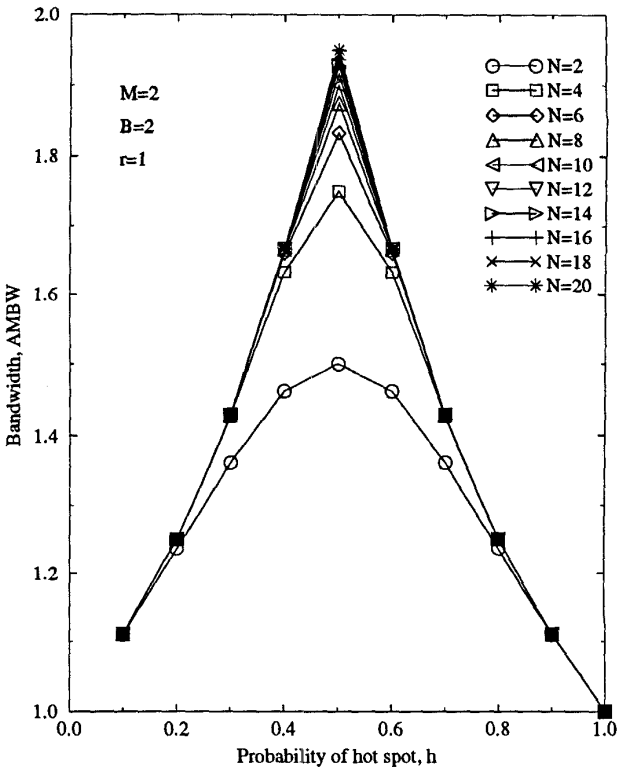

Figure 6: Bandwidth vs. hot spot probability for a N processors 2 memories system for $r = 1$

The bandwidth is upper bounded by two due to the number of processors being two. The bandwidth for a system consisting of two memories and two busses for $r = 1$ is shown in Figure 6. Average bandwidth is shown as a function of the hot spot probability for different number of processors. In such a system, $h < 0.5$ or $h > 0.5$ represent M_1 or M_0 being the HMs respectively. $h = 0.5$ is the uniform memory reference case. Therefore, the bandwidth falls off rapidly on either sides of $h = 0.5$ and the maximum bandwidth is obtained for $h = 0.5$. Increasing the number of processors increases the number of requests, resulting in a higher bandwidth. The increase in bandwidth is not significant for $N > 6$ because the memories are continuously busy and an increased number of requests from the processors can not be satisfied with only two memories. The number of memories have to be increased to increase the bandwidth.

Bandwidth for a system having 10 processors, 10 memories, and five busses is shown in Figure 7 as a function of the request rate for different hot spot probabilities. For uniform memory reference ($h = 0.1$), the bandwidth increases with an increase in the processor request rate. The bandwidth is limited to five by the five busses used. Doubling the number of busses would only increase the bandwidth by 5% for the case of $h = 0.1$ [3]. As the hot spot probability increases, the maximum achievable bandwidth decreases due to increased contention at the HM. For example, the bandwidth for $h = 0.5$ is limited to approximately two. For $h = 1$, all memory requests are directed to the HM, and the bandwidth is limited to one.

5 Conclusions

Modeling permits a fast and inexpensive method to evaluate the performance of multiprocessor systems. It allows the designer to choose appropriate system parameters at the design stage. Most of the previous models of multiple bus systems are based on the assumption of uniform memory reference pattern. We have developed Markov chain-based analytical models to evaluate the bandwidth of a multiple bus system in the presence of a single memory hot spot in the system. We have also removed the assumption of temporal independence of requests used by many authors. We have further shown that the number of states in a Markov chain model increases rapidly when the number of processors and memories is large in a hot spot case.

References

[1] Q. Yang, "Effects of arbitration protocols on the performance of multiple-bus multiprocessors," *1991 International Conference on Parallel Processing*, pp. I 600–603, August 12-16, 1991.

[2] Y.C. Liu and C.J. Jou, "Effective memory bandwidth and processor blocking probability in multiple-bus systems," *IEEE Transactions on Computers*, vol. C-36, no. 6, pp. 761–764, June 1987.

[3] T. Lang, M. Valero, and I. Alegre, "Bandwidth of crossbar and multiple-bus connections for multiprocessors,"

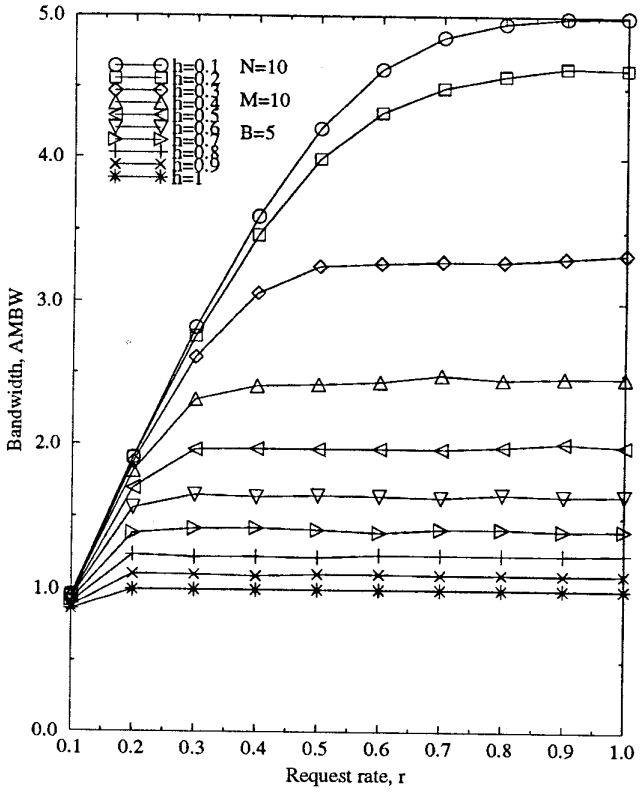

Figure 7: Bandwidth vs. request rate for a 10 processors and 2 memories system.

IEEE Transactions on Computers, vol. C-31, no. 12, pp. 1227–1234, December 1982.

[4] L.N. Bhuyan, "An analysis of processor-memory interconnection networks," *IEEE Transactions on Computers*, vol. C-34, no. 3, pp. 279–283, March 1985.

[5] B. Wilkinson, "Overlapping connectivity interconnection networks for shared memory mulliprocessor systems," *Journal of Parallel and Distributed Computing*, vol. 15, no. 1, pp. 49–61, May 1992.

[6] Y.C. Liu and C. Wang, "Analysis of prioritized crossbar multiprocessor systems," *Journal of Parallel and Distributed Computing*, vol. 7, pp. 321–334, 1991.

[7] D.W.L. Yen, J.H. Patel, and E.S. Davidson, "Memory interference in synchronous multiprocessor systems," *IEEE Transactions on Computers*, vol. C-31, no. 11, pp. 1116–1121, November 1982.

[8] K.O. Siomalas and B.A. Bowen, "Performance of crossbar multiprocessor systems," *IEEE Transactions on Computers*, vol. C-32, no. 7, pp. 689–695, July 1983.

[9] A.S. Sethi and N. Deo, "Interference in multiprocessor systems with localized memory access probabilities," *IEEE Transactions on Computers*, vol. C-28, no. 2, pp. 157–163, February 1979.

[10] M. Valero, J.M. Llaberia, J. Labarta, E. Sanvicente, and T. Lang, "A performance evaluation of the multiple bus network for multiprocessors," *Sigmetrics Conference on Measurement and Modelling of Computer Systems*, pp. 200–206, August 1983.

[11] D.P. Bhandarkar, "Analysis of memory interference in multiprocessors," *IEEE Transactions on Computers*, vol. C-24, no. 9, pp. 897–908, September 1975.

[12] C.H. Hoogendoorn, "A general model for memory interference in multiprocessors," *IEEE Transactions on Computers*, vol. C-26, no. 10, pp. 998–1005, October 1977.

[13] T.N. Mudge, J.P. Hayes, and D.C. Winsor, "Multiple-bus architectures," *Computer*, vol. 20, pp. 42–48, June 1987.

[14] Q. Yang and L.N. Bhuyan, "Analysis of packet-switched multiple-bus multiprocessors," *IEEE Transactions on Computers*, vol. 40, no. 3, pp. 352–357, March 1991.

[15] T.N. Mudge and H.B. Al-Sadoun, "A semi-markov model for the performance of multiple-bus systems," *IEEE Transactions on Computers*, vol. C-34, no. 10, pp. 934–942, October 1985.

[16] T.N. Mudge, J.P. Hayes, G.D. Buzzard, and D.C. Winsor, "Analysis of multiple-bus interconnection networks," *Journal of Parallel and Distributed Computing*, vol. 3, pp. 328–343, 1986.

[17] A. Fukuda, "Equilibrium point analysis of memory interference in multiprocessor systems," *IEEE Transactions on Computers*, vol. 37, no. 5, pp. 585–593, May 1988.

[18] D. Towsley, "Approximate models of multiple bus multiprocessor systems," *IEEE Transactions on Computers*, vol. C-35, no. 3, pp. 220–228, March 1986.

[19] A.K. Ramani, N. Gore, P.C. Sharma, and P.K. Chande, "Performance analysis of hierarchicaly structured multiple bus multiprocessor systems," *2nd IEEE Symposium on Parallel and Distributed Processing*, pp. 760–767, December 9-13, 1990.

[20] Y.H. Lee, S.E. Cheung, and J.K. Peir, "Consecutive requests traffic model in multistage interconnection networks," *International Conference on Parallel Processing*, pp. I 534–541, August 12-16, 1991.

[21] K.B. Irani and I.H. Onyuksel, "A closed-form solution for the performance analysis of multiple-bus multiprocessor systems," *IEEE Transactions on Computers*, vol. C-33, no. 11, pp. 1004–1012, November 1984.

[22] G.F. Pfister and V.A. Norton, "Hot spot contention and combining in multistage interconnection networks," *IEEE Transactions on Computers*, vol. C-34, no. 10, pp. 943–948, October 1985.

[23] H.S. Kim and A. Leon-Garcia, "Performance of buffered Banyan networks under non-uniform traffic patterns," *IEEE Transactions on Communications*, vol. 38, no. 5, pp. 648–658, May 1990.

[24] T. Lin and L. Kleinrock, "Performance analysis of finite-buffered multistage interconnection networks with a general traffic pattern," *1991 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, San Diego, CA, pp. 68–78, May 21-24, 1991.

[25] M. Atiquzzaman and M.S. Akhtar, "Performance of buffered multistage interconnection networks in non uniform traffic environment," *7th International Parallel Processing Symposium*, California, pp. 762–767, April 13-16, 1993.

[26] M. Atiquzzaman and M.S. Akhtar, "Effect of hot spots on the performance of multistage interconnection networks," *FRONTIERS 92: The Fourth Symposium on the Frontiers of Massively Parallel Computation*, Virginia, pp. 504–505, October 19-21, 1992.

[27] M. Atiquzzaman and M.S. Akhtar, "Effect of non-uniform traffic on the performance of unbuffered multistage interconnection networks," *IEE Proceedings Part-E*, To appear in 1994.

[28] B. Zhou and M. Atiquzzaman, "Performance of output-multibuffered multistage interconnection networks under nonuniform traffic patterns," *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'94)*, North Carolina, USA, January 31 - February 2, 1994.

[29] W.T. Chen and J.P. Sheu, "Performance analysis of multiple bus interconnection networks with hierarchical requesting model," *IEEE Transactions on Computers*, vol. 40, no. 7, pp. 834–842, July 1991.

[30] M.A. Sayeed and M. Atiquzzaman, "Peformance of multiple-bus mulitprocessor under non-uniform memory reference model," *19th International Symposium on Computer Architecture*, Gold Coast, Australia, p. 432, May 19-21, 1992.

[31] M. Atiquzzaman and M.M. Banat, "Effect of hot-spots on the performance of crossbar multiprocessor systems," *Parallel Computing*, vol. 19, no. 4, pp. 455–461, April 1993.

[32] K.P. Bogart, *Introductory Combinatorics*. Harcourt Brace Jovanovich, 2nd ed., 1990.