



Effect of non-uniform memory request on the performance of buffered multiprocessor systems

Mohammed Azhar Sayeed^a, Mohammed Atiquzzaman^{b,*}

^a Cisco Systems Inc., 250 Apollo Drive, Chelmsford, MA 01824, USA

^b Department of Electrical and Computer Engineering, University of Dayton, 300 College Park, Dayton, OH 45469-0226, USA

Received 29 March 2000; accepted 12 April 2000

Abstract

Performance analysis of multiple-bus systems is usually carried out under the assumption of a uniform memory request model. Hot spots arising in multiprocessor systems give rise to non-uniform memory requests. It is known that a hot spot memory request 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*, and to compare the performance of buffered and unbuffered systems. Analytical models based on Markov chains have been developed to determine the bandwidth of buffered multiple-bus systems in the presence of a hot spot memory request 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 to evaluate the performance of a buffered multiple-bus system in the presence of a hot spot memory request pattern. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Multiprocessor systems; Performance analysis; Markov chain modeling; Non-uniform memory request

1. Introduction

A multiprocessor system essentially consists of a number of processors and memories interconnected by an interconnection network. Networks have topologies such as multiple-bus, multistage, hypercube, tree, etc. Multiple-bus systems have been found to be suitable for small and medium sized systems. Multiple-bus systems are modular, easily expandable and fault tolerant. However, the performance of multiple-bus systems is limited due to bus and memory contention.

* Corresponding author. Tel.: +1-937-229-3183; fax: +1-937-229-4529.

E-mail addresses: asayeed@cisco.com (M.A. Sayeed), atiq@ieee.org (M. Atiquzzaman).

Different conflict resolution strategies are used to manage such contention [1,2]. Commonly used criteria for the performance measurement of multiple-bus systems include bandwidth, probability of request acceptance, and processor utilization.

Multiple-bus systems can be buffered or unbuffered. In an unbuffered system, unsuccessful processor requests (due to contention) during a cycle are discarded. The processors have to re-submit 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 [3–9] and buffered [10,11] crossbar and multiple-bus systems have been reported in the literature. Good surveys of crossbar and multiple-bus systems appear in Refs. [12,13].

In evaluating the performance of crossbar or multiple-bus systems, most authors assume a uniformly distributed memory request pattern, whereby a memory request generated by a processor is equally likely to be directed to any of the memory modules [6,14–19]. However, the uniformly distributed memory request pattern is rather restrictive in real-world situations. Studies have shown that non-uniform memory request patterns may arise in multiprocessor systems. Favorite memory request pattern, a particular type of non-uniform traffic pattern, has been analyzed for multiple-bus systems in Ref. [20], while a consecutive request pattern has been studied in Ref. [21]. Successive requests with local referencing has been discussed in Ref. [22]. A hot spot request pattern in multistage interconnection networks have been studied in Refs. [23–27]. Other non-uniform request patterns can be found in Ref. [28].

One of the reasons for hot spot memory request patterns 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 location, thereby giving rise to a hot spot memory request pattern in the system. Combining has been shown to alleviate, the problem arising due to a hot spot in multistage interconnection networks in the case of the hot spot being 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 request pattern.

Performance evaluation of unbuffered multiple-bus and crossbar systems under a hot spot request pattern has been reported in Refs. [29,30], respectively. It was found that the performance degrades significantly with an increase in the proportion of hot spot requests. The objectives of this paper are to

- develop analytical models to *study* the performance of buffered multiple-bus systems in the presence of a hot spot memory request pattern,
- *compare* the performance of buffered and unbuffered systems.

The commonly used temporal assumption of processor requests in consecutive memory cycles is not realistic because a processor will not issue another request until the previous request has been satisfied. The model removes the above temporal assumption of processor requests in consecutive memory cycles. 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 those cases where the state space of the Markov 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 and compared in Section 4, and concluding remarks appear in Section 5.

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 are used in the Markov chain model described in Section 3:

- The system (Fig. 1) consists of a multiple-bus system with N processors (P_0, P_1, \dots, P_{N-1}) M memory modules (M_0, M_1, \dots, 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., a request generated by a processor during a cycle is independent of requests generated by other processors during the same cycle.
- A processor having an outstanding request is *blocked*, and cannot 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 . We call this the *static request rate*. It will be

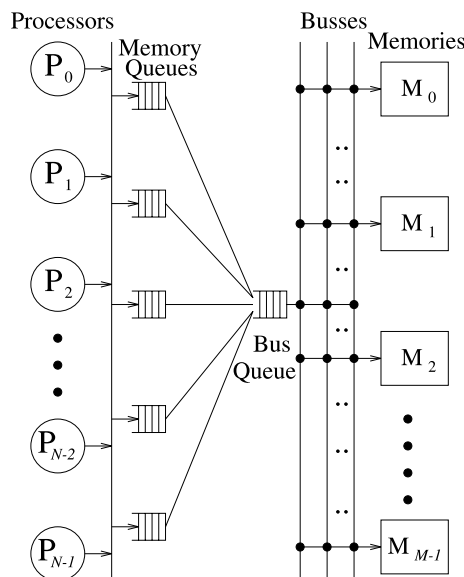


Fig. 1. A queued multiple-bus system.

shown in Section 4 that the effective request rate of a processor is significantly less than r in the case of high blocking.

- The probability that the generated request is for the hot memory module (HM) or a non-hot memory module (NHM) is h or $\bar{h} = (1 - h)/(M - 1)$, 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, with outstanding requests, services one request from its buffer.
- Processor requests which cannot be served during the same cycle as they were generated remain queued at the buffers for servicing at a later cycle. The buffers are considered to be of *infinite size*.
- In the case of bus conflicts, memory requests to be serviced are chosen at random from the outstanding requests. A total of B requests are presented to B different memory modules.

3. Performance evaluation

In this section, we develop a Markov chain based analytical model for the performance evaluation of a buffered multiple-bus system operating according to the assumptions described in Section 2. It is well known that Markov chain modeling of an $N \times M$ system, even for a uniform memory request pattern, results in a large number of states, which makes it analytically intractable [31,32]. In the presence of a hot spot memory request pattern, the number of states is even larger. Therefore, for a multiple-bus system in the presence of a hot spot memory request pattern, we develop Markov chain models for systems having a small number of processors or memories [31]. 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. In the next few sections, we separately model systems having large number of processors or memories. We then show that the model, for systems having a large number of processors and memories, is too complex to be solved with reasonable efforts.

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 $(S_0, S_1, S_2, \dots, S_{M-1})$, 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 reduce the states of the system to seven equivalent states (Fig. 2), $\tilde{\pi}_{2,M} = \{\tilde{\pi}_0, \tilde{\pi}_1, \tilde{\pi}_2, \dots, \tilde{\pi}_6\}$, which are based on the number of outstanding requests for the HM and NHMs. The equivalent states are as follows:

- $\tilde{\pi}_0 = (2,0)$: two outstanding requests for the HM,
- $\tilde{\pi}_1 = (1,0)$: one outstanding request for the HM,
- $\tilde{\pi}_2 = (1,1)$: one outstanding request for the HM and one for a NHM,
- $\tilde{\pi}_3 = (0,0)$: no outstanding requests,
- $\tilde{\pi}_4 = (0,1,1)$: two outstanding requests for two different NHMs,

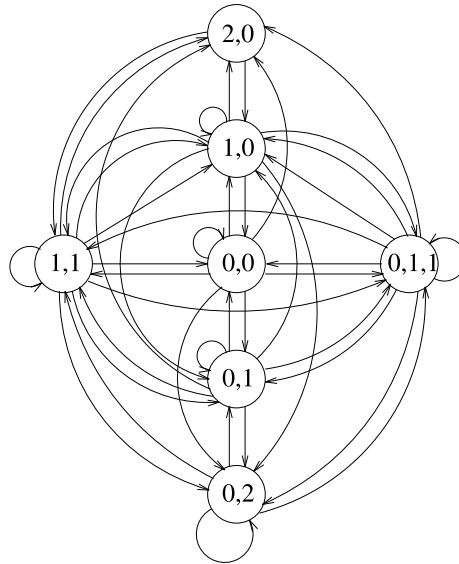


Fig. 2. Markov diagram of a 2 processors, M memories system for $r < 1$.

- $\tilde{\pi}_5 = (0,1)$: one outstanding request for a NHM,
- $\tilde{\pi}_6 = (0,2)$: 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 $\tilde{\pi}_6$ is composed of the following states: $(0, 2, 0, \dots, 0), (0, 0, 2, 0, \dots, 0), (0, 0, 0, 2, 0, \dots, 0), \dots (0, 0, \dots, 0, 2)$.

Having defined the states of the system, the transition probabilities between the states will be determined next. The transition probabilities, $P_{2,M} = \{p_{i,j}, 0 \leq i, j \leq 6\}$, for a $2 \times M$ system will be represented by a 7×7 matrix. The state transition from state i to state j is therefore, represented by $p_{i,j}$. The Markov chain, with all the possible transitions, is shown in Fig. 2. Each arc represents a transition from one state to another. When the system is in state $\tilde{\pi}_0$, 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 $p_{0,1} = (1 - h)$. Other transition probabilities can be derived similarly. The transition probability matrix $P_{2,M}$ is therefore, given by

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

It is possible to make a transition from any state, back to the same state in a finite number of transitions, and hence the chain is aperiodic. Since the chain is also irreducible, it is ergodic and hence possesses a unique stationary probability distribution of the states. Let the stationary probability distribution of states $\tilde{\pi}_{2,M} = \{\tilde{\pi}_0, \tilde{\pi}_1, \tilde{\pi}_2, \tilde{\pi}_3, \tilde{\pi}_4, \tilde{\pi}_5, \tilde{\pi}_6\}$, be represented by $\pi_{2,M} = \{\pi_0, \pi_1, \pi_2, \dots, \pi_6\}$. We solve the following two equations:

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

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

to obtain the stationary probability distributions $\pi_{2,M} = \{\pi_0, \pi_1, \dots, \pi_6\}$. Solution of the equations gives

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

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

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

where (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, for $r = 1$, a model to determine the average memory bandwidth of a system having N processors, two memories, and two busses. Since there are only two memories, the state of the system can be described by a two-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 $\pi_{4,2} = \{(4, 0), (3, 1), (2, 2), (1, 3), \text{ and } (0, 4)\}$ and a 9×2 system has the states $\pi_{9,2} = \{(9, 0), (8, 1), (7, 2), (6, 3), (5, 4), (4, 5), (3, 6), (2, 7), (1, 8), (0, 9)\}$. 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 Fig. 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 proba-

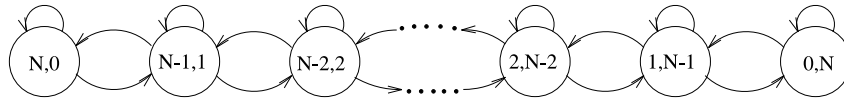


Fig. 3. Markov diagram of an N processors, two-memory system.

bilities $p_{0,0} = h$ 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 only one request is serviced. Therefore, $\mu_0 = \mu_N = 1$ and $\mu_i = 2, 1 \leq i \leq N - 1$. The transition probability matrix for the Markov chain in Fig. 3 is therefore, given by

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

The Markov chain is ergodic, and thus a stationary probability distribution $\pi_{N,2} = \{\pi_0, \pi_1, \dots, \pi_N\}$ exists. The equations describing the Markov chain of an $N \times 2$ system are given by

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

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

Solving Eqs. (4) and (5) yields

$$\pi_0 = \frac{\alpha - 1}{\alpha^{2N} - 1},$$

$$\pi_1 = \alpha(\alpha + 1) \frac{\alpha - 1}{\alpha^{2N} - 1},$$

⋮

$$\pi_i = \alpha^2 \pi_{i-1}, \quad 2 \leq i \leq N - 1,$$

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

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

to obtain

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

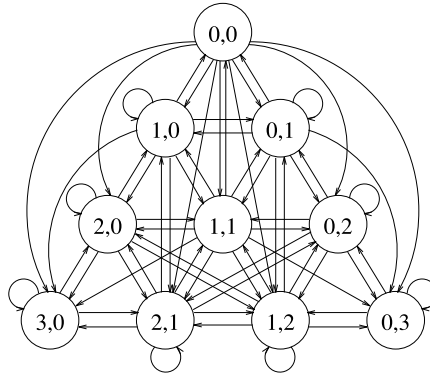


Fig. 4. Markov chain of a system having three processors and two memories, for $r < 1$.

The AMBW for $r = 1$ is obtained as described above. Because of the large number of possible states, the Markov chain for $r < 1$ is too complex for an $N \times 2$ system. To illustrate the complexity of the chain, the number of transitions of the chain for a 3×2 system, for $r < 1$, is shown in Fig. 4. Consequently, we are forced to obtain the results for $r < 1$ case using simulations.

3.3. Modeling an $N \times M$ system

Consider first a 4×4 system under the uniform memory request 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 request pattern, the total number of partitions of N into M parts is given [33] by $\sum_{n=1}^M \mathcal{P}(N, n)$, where $\mathcal{P}(N, n)$ is given by the recurrence relation

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

where, $\mathcal{P}(N, 1) = \mathcal{P}(N, N) = 1$. The N requests from the N processors can go to any number of memory modules within the M memory modules. If we denote the system state in a decreasing list format, then such a list (i_1, i_2, \dots, i_M) , $i_1 \geq i_2 \geq i_3 \dots \geq i_M$, 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 *uniformly distributed* memory request pattern.

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

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

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 state probability distribution. Simulation was therefore, used to determine the bandwidth of $N \times M$ systems, where N and M are greater than 2.

4. Results

In this section, we present performance figures for a buffered system in the presence of a hot spot request pattern. We also compare the performance of a buffered system with that of an unbuffered system. Results for an unbuffered system are obtained from the models developed in Ref. [34]. Results obtained for $2 \times M$, $N \times 2$, and $N \times M$ multiple-bus systems using the models and simulators described in the previous sections are shown in Figs. 5–7.

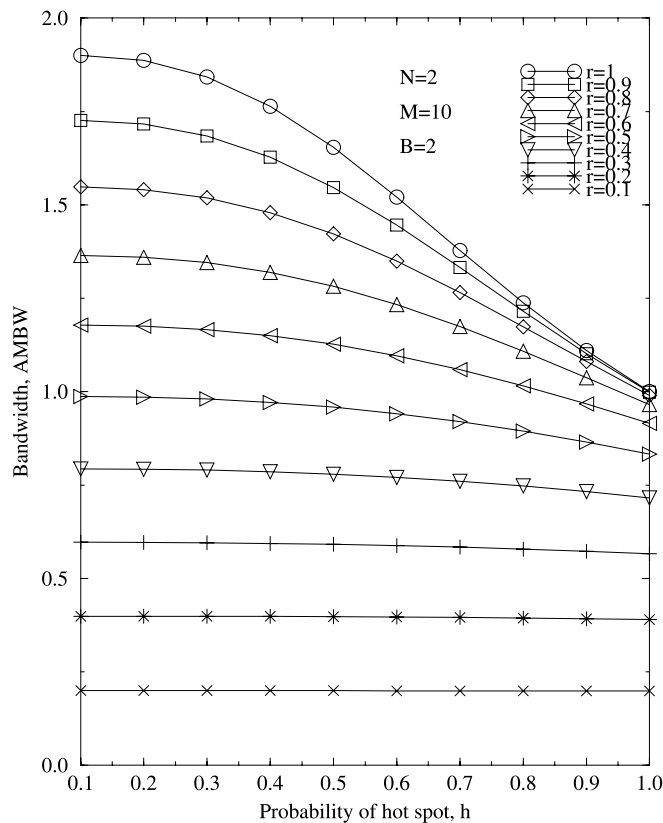


Fig. 5. Bandwidth vs. hot spot probability for systems with two processors, 10 memories, two busses and $r < 1$.

Fig. 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 and for different request rates. For a 2×10 system, $h = 0.1$ corresponds to the uniform memory request 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 a small contention for the HM at low values of processor request rates. It should be noted that processors with blocked requests cannot 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 for preventing the buffers from overflowing.

Due to the number of processors being two, the upper bound of the bandwidth is two. The bandwidth for a system consisting of two memories and two busses, for $r = 1$, is shown in Fig. 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 and $h = 0.5$ is the uniform memory request case. Therefore, the bandwidth falls off rapidly on either side of $h = 0.5$, and the maximum bandwidth is obtained for $h = 0.5$. Increasing the number

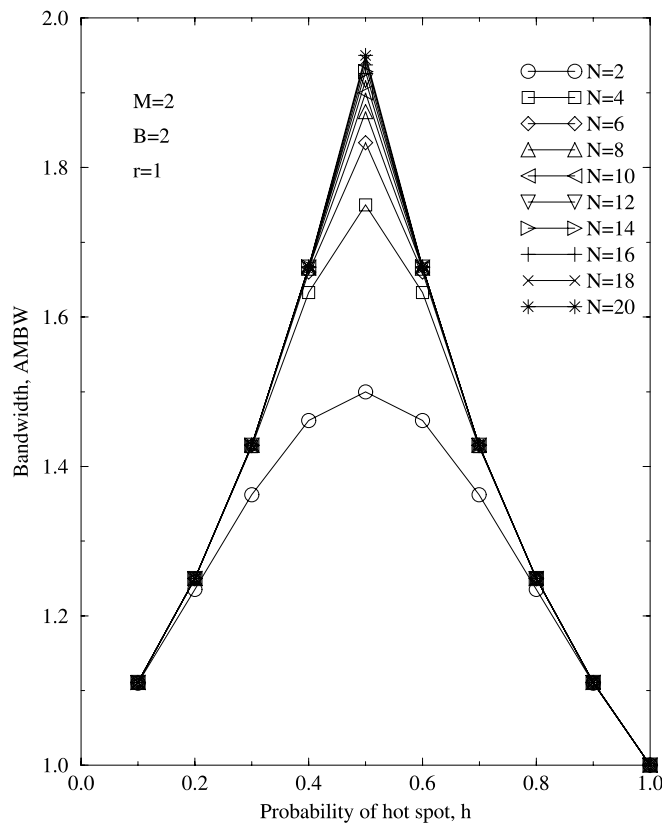


Fig. 6. Bandwidth vs. hot spot probability for systems with N processors, two memories, two busses and $r = 1$.

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 cannot be satisfied with only two memories. The number of memories have to be increased to achieve an increase in the bandwidth with an increasing number of processors.

Bandwidth for a system having 10 processors, 10 memories, and five busses is shown in Fig. 7 as a function of the request rate for different hot spot probabilities. For uniform memory request ($h = 0.1$), 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, thereby making the system a crossbar, would only increase the bandwidth by 5% for the case of $h = 0.1$ [35]. 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 due to memory contention at the HM. For $h = 1$, all memory requests are directed to the HM, and the bandwidth is limited to one.

Unbuffered multiple-bus systems, where the rejected requests are simply dropped, have been studied in the presence of a hot spot request pattern in Ref. [34]. A comparison of the bandwidths of unbuffered and buffered multiple-bus systems is shown in Fig. 8 for a system having 10

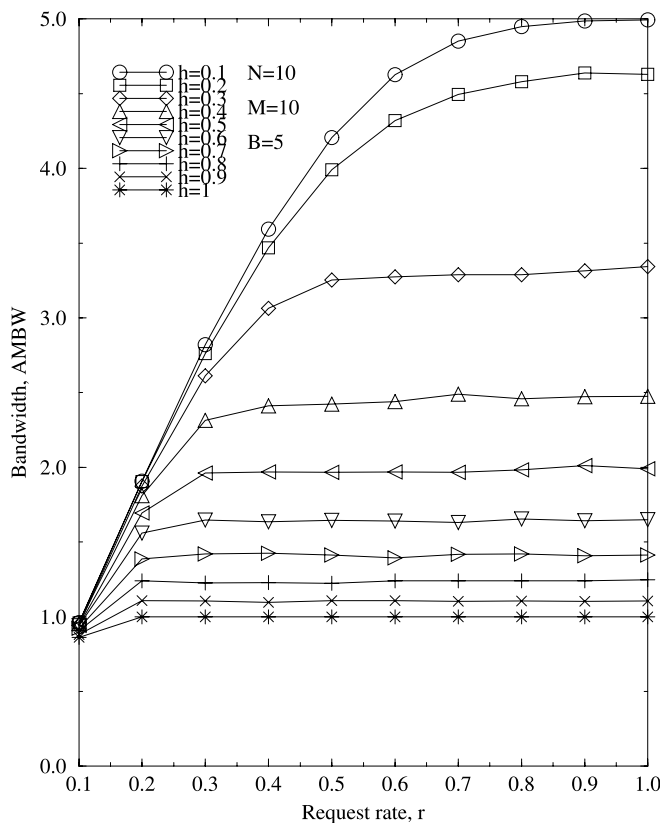


Fig. 7. Bandwidth vs. request rate for a system with 10 processors, two memories and five busses.

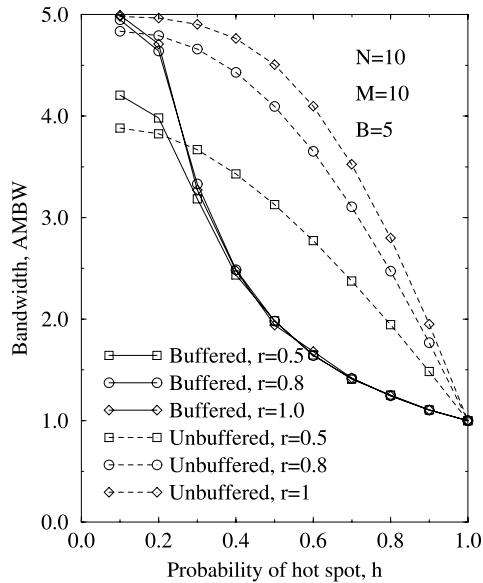


Fig. 8. A comparison of the bandwidths of buffered and unbuffered systems with 10 processors, 10 memories and five busses.

processors, 10 memories, and five busses. The buffered system has a higher bandwidth only in the case of a uniform memory request ($h = 0.1$). As the hot spot probability increases, the bandwidth of the buffered system falls rapidly. The reason is that processors in a buffered system remain blocked until they receive service from the memory modules. Blocked processors cannot generate requests, resulting in a much lower *effective request rate*. For $h > 0.2$, the queue lengths become large, resulting in a sharp drop in the bandwidth. Since requests are discarded in an unbuffered system, the effective request rate in such a system is the same as the static request rate (r). On the other hand, the effective request rate of a buffered system is given by BW/N . For example, we find in Fig. 8 that a buffered system has an effective request rate of $2.8/10 = 0.28$ for $h = 0.8$ and a static request rate of 1.0. This illustrates the difference in the static and effective request rates for a buffered system, and hence accounts for a lower bandwidth of a buffered system when compared to an unbuffered system having the same static request rates. Due to the dropping of requests, an unbuffered system in fact, gives an optimistic view of the memory bandwidth.

To show that a buffered system has a lower bandwidth than an unbuffered system due to a reduction in the effective request rate of a buffered system, we have compared the bandwidths of a buffered and an unbuffered system for low hot spot probabilities and low request rates in Fig. 9. It is seen that for a uniform traffic ($h = 0.1$), the bandwidth of a buffered system is always higher than an unbuffered system. For hot spot probabilities of 0.2 and 0.3, the buffered system is better than an unbuffered system when the request rate is less than 0.63 and 0.35, respectively. Since higher request rate and/or higher hot spot probability results in blocking in a buffered system (thereby resulting in a drop in the effective request rate), the bandwidth of a buffered system in such a case is lower than that of an unbuffered system.

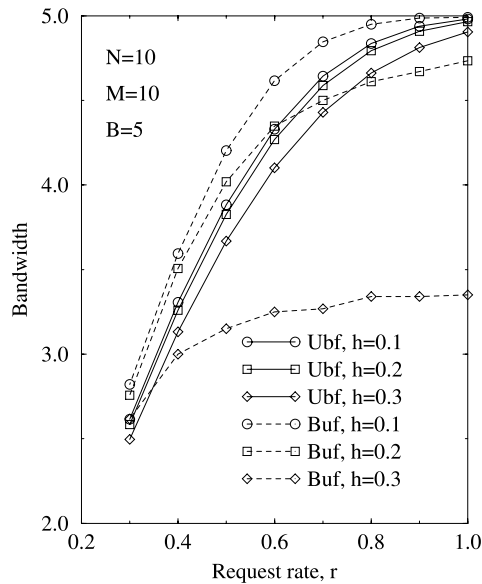


Fig. 9. A comparison of the bandwidths of a buffered and unbuffered system for different processor request rates and low hot spot probabilities. $N = 10$, $M = 10$, and $B = 5$.

The analytical model presented in Section 4 has been validated with results obtained from a stochastic simulator. The simulator was driven by a hot spot memory request pattern, and the number of memories accessed per memory cycle was observed for 50,000 memory cycles, the average of which gave the bandwidth. Results obtained from the analytical model were found to be in close agreement to the simulation results.

5. Conclusions

Analytical 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. The model helps in choosing the number of memory modules required to achieve a given bandwidth for a given number of processors and processor request rates. Most of the previous models of multiple-bus systems are based on the assumption of uniformly distributed memory request pattern. We have developed Markov chain-based analytical models to evaluate the performance of a multiple-bus system in the presence of a single memory hot spot in the system. The average memory bandwidth was taken as the performance measure. We have also *removed the assumption of temporal independence of requests* used by most researchers. We have illustrated that the number of states in a Markov chain model increases rapidly when the number of processors and memories is large in a multiple-bus system having non-uniform request pattern. Computer simulations have been used to validate our analytical models. The proposed model can be used to evaluate the performance of multiple-bus systems in the presence of other non-uniform traffic, like the favorite memory, hierarchical request pattern, etc.

The bandwidth of a buffered system has been compared with that of an unbuffered system in the presence of a non-uniform request pattern. Since processors with outstanding requests in a buffered system cannot generate a new request, it was shown that the effective request rate decreases drastically from the static request rate. On the contrary, since the rejected requests are simply dropped in the case of an unbuffered system, the effective request rate is the same as the static request rate in such a system.

References

- [1] Chung CM, Chiang DA, Yang Q. Comparative analysis of different arbitration protocols for multiple-bus multiprocessors. *J Comput Sci Tech* 1996;11(3):313–25.
- [2] Yang Q. Effects of arbitration protocols on the performance of multiple-bus multiprocessors. 1991 International Conference on Parallel Processing. August 12–16, 1991. p. I 600–603.
- [3] Ku HK, Hayes JP. Connective fault tolerance in multiple-bus systems. *IEEE Trans Parall Distrib Sys* 1998;8(6):574–86.
- [4] Sayeed MA, Atiquzzaman M. Performance of multiple bus multiprocessor system under hot spots. *Comput Electr Engng J* 1999;25(2):77–94.
- [5] Mahmud SM, Samaratinga LT, Munteanu MD. Multiple bus-based hierarchical multiprocessors and their bandwidth analysis. *IEEE International Conference on Algorithms and Architectures for Parallel Processing*, Singapore, 1996. p. 311–318.
- [6] Liu YC, Jou CJ. Effective memory bandwidth and processor blocking probability in multiple-bus systems. *IEEE Trans Comput* 1987;C-36(6):761–4.
- [7] Chaudhry GM, Khan AN. Bandwidth of a reconfigurable multiple-group multiprocessor system. *J Sys Arch* 1996;24(3):225–34.
- [8] Wilkinson B. Overlapping connectivity interconnection networks for shared memory multiprocessor systems. *J Parall Distrib Comput* 1992;15(1):49–61.
- [9] Liu YC, Wang C. Analysis of prioritized crossbar multiprocessor systems. *J Parall Distrib Comput* 1991;7: 321–34.
- [10] Valero M, Llaberia JM, Labarta J, Sanvicente E, Lang T. A performance evaluation of the multiple bus network for multiprocessors. *Sigmetrics Conference on Measurement and Modelling of Computer Systems* August 1983. p. 200–206.
- [11] Wang JH, Wu HZ, Jiang YD. Performance analysis of prioritized multiple-bus multiprocessor systems. *Int J Mini Multicomp* 1996;18(1):46–50.
- [12] Youn HY, Chen CCY. A comprehensive performance evaluation of crossbar networks. *IEEE Trans Parall Distrib Sys* 1993;4(5):481–9.
- [13] Mudge TN, Hayes JP, Winsor DC. Multiple-bus architectures. *Computer* 1987;20:42–8.
- [14] Yang Q, Bhuyan LN. Analysis of packet-switched multiple-bus multiprocessors. *IEEE Trans Comput* 1991;40(3):352–7.
- [15] Mudge TN, Al-Sadoun HB. A semi-markov model for the performance of multiple-bus systems. *IEEE Trans Comput* 1985;C-34(10):934–42.
- [16] Mudge TN, Hayes JP, Buzzard GD, Winsor DC. Analysis of multiple-bus interconnection networks. *J Parall Distrib Comput* 1986;3:328–43.
- [17] Fukuda A. Equilibrium point analysis of memory interference in multiprocessor systems. *IEEE Trans Comput* 1988;37(5):585–93.
- [18] Towsley D. Approximate models of multiple bus multiprocessor systems. *IEEE Trans Comput* 1986;C-35(3): 220–8.

- [19] Ramani AK, Gore N, Sharma PC, Chande, PK. Performance analysis of hierarchically structured multiple bus multiprocessor systems. Second IEEE Symposium on Parallel and Distributed Processing, December 9–13, 1990. p. 760–767.
- [20] Bhuyan LN. An analysis of processor-memory interconnection networks. *IEEE Trans Comput* 1985;C-34(3):279–83.
- [21] Lee YH, Cheung SE, Peir JK. Consecutive requests traffic model in multistage interconnection networks. International Conference on Parallel Processing, August 12–16, 1991. p. I 534–541.
- [22] Irani KB, Onyuksel IH. A closed-form solution for the performance analysis of multiple-bus multiprocessor systems. *IEEE Trans Comput* 1984;C-33(11):1004–12.
- [23] Kim HS, Leon-Garcia A. Performance of buffered Banyan networks under non-uniform traffic patterns. *IEEE Trans Commun* 1990;38(5):648–58.
- [24] Lin T, Kleinrock L. 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, May 21–24, 1991. p. 68–78.
- [25] Atiquzzaman M, Akhtar MS. Performance of buffered multistage interconnection networks in non-uniform traffic environment. Seventh International Parallel Processing Symposium, California, April 13–16, 1993. p. 762–767.
- [26] Atiquzzaman M, Akhtar MS. Effect of hot spots on the performance of multistage interconnection networks. *FRONTIERS 92: The Fourth Symposium on the Frontiers of Massively Parallel Computation*, Virginia, October 19–21, 1992. p. 504–505.
- [27] Atiquzzaman M, Akhtar MS. Effect of non-uniform traffic on the performance of unbuffered multistage interconnection networks. *IEE Proc – Comput Digi Tech* 1994;141(3):169–76.
- [28] Chen DX, Mark JW. Performance analysis of output buffered fast packet switches with bursty traffic loading. *Globecom 91: IEEE Global Telecommunications Conference*, Arizona, December 2–5, 1991. p. 455–459.
- [29] Sayeed MA, Atiquzzaman M. Performance of multiple-bus multiprocessor under non-uniform memory reference model. 19th International Symposium on Computer Architecture, Gold Coast, Australia, May 19–21, 1992. p. 432.
- [30] Atiquzzaman M, Banat MM. Effect of hot-spots on the performance of crossbar multiprocessor systems. *Parall Comput* 1993;19(4):455–61.
- [31] Sethi AS, Deo N. Interference in multiprocessor systems with localized memory access probabilities. *IEEE Trans Comput* 1979;C-28(2):157–63.
- [32] Bhandarkar DP. Analysis of memory interference in multiprocessors. *IEEE Trans Comput* 1975;C-24(9): 897–908.
- [33] Bogart KP. *Introductory Combinatorics*. 2nd ed. Harcourt Brace Jovanovich: New York, 1990.
- [34] Sayeed MA, Atiquzzaman M. Performance of multiple-bus multiprocessor under non-uniform memory reference model. International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'94), North Carolina, USA, January 31–February 2, 1994. p. 169–173.
- [35] Lang T, Valero M, Alegre I. Bandwidth of crossbar and multiple-bus connections for multiprocessors. *IEEE Trans Comput* 1982;C-31(12):1227–34.



Mohammed Azhar Sayeed is working as a Senior Product Manager, Unified Communications in the ITD (Internet Technologies Division) with Cisco Systems, responsible for product management and rolled out of unified communications solutions in Cisco software. Prior to working for Cisco he worked at Cabletron Systems as a Senior Product Manager and later as an ATM Marketing Manager. He started his networking career as field service engineer working on installations in X.25 and Frame Relay. He has over nine years of experience in the networking industry and has designed, implemented and troubleshot LANs and WANs using multivendor gear. Prior to working with Cabletron he was at Digital Equipment Corporation as an ATM Technical Marketing Engineer (Aviator). He has represented Digital, Cabletron and now Cisco as a speaker at major conferences such as ATM Year, Integrated Broadband Networks and Comdex. His research interests include QoS for IP telephony, Interdomain QoS and IP telephony architecture and protocols. He can be reached at asayeed@cisco.com



Mohammed Atiquzzaman received the M.Sc and Ph.D. degrees in electrical engineering and electronics from the University of Manchester Institute of Science and Technology, England in 1984 and 1987, respectively. Currently, he is a faculty member in the department of Electrical and Computer Engineering at University of Dayton, OH. He serves on the editorial boards of *IEEE Communications Magazine*, *Computer Communications* journal, *Telecommunication Systems* journal and *Real Time Imaging*. He has guest edited 10 special issues of various journals including “Switching and Traffic Management for Multimedia” and “Optical Networks, Systems and Devices” of the *IEEE Communications Magazine*, “Internet of the Future: Architectures and Protocols” of the *European Transactions on Telecommunications*, “ATM Switching” and “ATM Networks” of the *International Journal of Computer Systems Science and Engineering*, special issue on Projection-based Transforms in the *Image and Vision Computing* journal. He has also served in the technical program committee of many national and international conferences including *IEEE INFOCOM* and *IEEE Annual Conference on Local Computer Networks*. His current research interests are in Broadband ISDN and ATM networks, multiprocessor systems, interconnection networks, parallel processing and image processing. He has over 90 refereed publications in the above areas. He can be contacted at atiq@ieee.org and his home page is at <http://www.engr.udayton.edu/faculty/matiquzz/>