

PARCO 734A

Short communication

Effect of hot-spots on the performance of crossbar multiprocessor systems

M. Atiquzzaman^a and M.M. Banat^b

^a Department of Computer Science & Computer Engineering, La Trobe University, Melbourne, Victoria 3083, Australia

^b Electrical Engineering Department, University of Ottawa, Ont., Canada K1N 6N5

Received 26 March 1992

Abstract

Atiquzzaman, M. and M.M. Banat, Effect of hot-spots on the performance of crossbar multiprocessor systems, *Parallel Computing* 19 (1993) 455–461.

Previous studies on the performance of crossbar multiprocessor systems have assumed a uniform memory reference model. Hot-spots arising in multiprocessor systems due to the use of shared variables, synchronization primitives, etc. give rise to nonuniform memory reference pattern. This paper presents an analytical expression for calculating the average memory bandwidth of a crossbar multiprocessor system having a hot memory. The expression is verified by simulation results.

Keywords. Average memory bandwidth; hot-spot memory contention; multiprocessors; performance evaluation; probabilistic analysis.

1. Introduction

Different types of interconnection networks exist to connect N processors to M memories in a multiprocessor system [1]. Because of modularity and fault tolerance of multiple-bus systems, such systems using B busses have been widely investigated [2]. Crossbar is a special case of multiple-bus systems where $B = \min(M, N)$. Contention in a crossbar system arises due to many processors trying to access common memory modules. The contention is usually measured by the bandwidth of the system. Both analytical modelling and simulation techniques have been used to determine the performance of crossbar systems [3,4]. Most of the authors have used the *uniform reference* model of memory reference in determining analytical expressions for performance [5–15].

Memory references in a multiprocessor system are not necessarily uniform. Non-uniformities arising due to locality of references have been analyzed. Das [16] and Bhuyan [17] have

Correspondence to: M. Atiquzzaman, Department of Computer Science & Computer Engineering, La Trobe University, Melbourne, Victoria 3083, Australia.

analyzed the case where each processor has a different favorite memory which it accesses with a higher probability than other memories. A model with local referencing, where the probability of successive requests to the same memory module is higher, has been analyzed by Sethi [18] and Irani [19]. This sort of referencing arises in cases like array accesses by user programs. Siomalas [20] has analyzed a system where each processor has a local reference pattern and a local memory module which it references immediately after referencing another memory module. This has been called the immediate return reference pattern. Mudge [21] has used a favorite memory assumption similar to that used in [16].

Non-uniform references may arise in a multiprocessor system due to variables used for locking, global and barrier synchronization, pointers to shared queues, etc. These are indivisible primitives, and must be stored in a single shared memory. The primitives will be accessed by all processors, giving rise to an increased reference to the memory modules where they are stored. This type of memory modules are called *hot memories*, and the phenomenon as *hot-spot contention* which was first noticed by Pfister [22]. In multistage interconnection networks, it gives rise to tree-saturation which severely degrades the performance of interconnection networks. Combining and feedback schemes have been suggested as solutions to the problem [22–24].

The objective of this paper is to determine the performance of crossbar multiprocessor systems under hot-spot conditions. One of the memories is considered a hot memory. An analytical expression for the average memory bandwidth will be derived using a probabilistic approach in Section 3. To evaluate the correctness of the analytical expressions, simulation results will be presented and compared with the results obtained from closed-form expressions.

2. Assumptions

In order to develop a closed-form solution, we model the multiprocessor system as follows.

- The system consists of N identical processors and M identical memories, $N \geq M$.
- The processors and memories are connected by a *cross-bar* switch (Fig. 1), i.e. the number of busses is equal to $\min(N, M)$. Contention in the system is due to memory conflicts only.
- The system operates *synchronously*, i.e. each processor generates a request at the start of a memory cycle resulting in a total number of N requests per cycle. All memory modules having at least one outstanding request are accessed simultaneously.

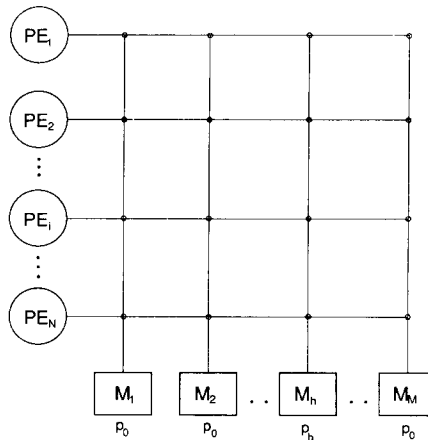


Fig. 1. A typical crossbar multiprocessor with hot memory.

- Processor requests are *spatially independent*, i.e. requests from the different processors are independent of each other.
- *Temporal independence* of requests is assumed, i.e. requests from a processor at different memory cycles are independent, implying that blocked requests are discarded.
- Processor requests to the different memory modules are *not uniformly distributed*. Memory module M_h is a hot memory for all the processors. The probability that PE $_i$, $i = 1, 2, \dots, N$ requests M_h is p_h and of requesting M_j , $j \neq h$ is given by $p_0 = (1 - p_h)/(M - 1)$, where $p_h > 1/M$ (if $p_h < 1/M$, then M_h becomes a disliked memory of all other processors).

3. Performance evaluation

In this section, the performance of the multiprocessor under the above assumptions will be determined. Average memory bandwidth will be used as the performance criteria. It is defined to be

$$\text{BW} = \sum_{k=1}^M k \cdot \text{Pr}(k), \quad (1)$$

where $\text{Pr}(k)$ is the probability that k memory modules are busy in a cycle. Let

R_k = event that k memory modules are accessed during a memory cycle.

$E_k^{(p,q)}$ = event that k memory modules are accessed during a cycle, and p and q are the number of references for M_h and M_j s, $j \neq h$ respectively. Hence, $p + q = N$.

R_k can be divided into two classes of events depending on the memory reference pattern:

Class 0: None of the k memories accessed during a cycle is the hot memory module M_h . This will result in the event $E_k^{(0,N)}$.

Class 1: One of the k memories accessed during a cycle is the hot memory module. This will result in the event $E_k^{(i,N-i)}$, $i \neq 0$.

The probability that k memory references are made during a cycle is therefore given by

$$\text{Pr}(R_k) = \text{Pr}(E_k^{(0,N)}) + \text{Pr}(E_k^{(i,N-i)}), \quad i \neq 0. \quad (2)$$

3.1 Determination of Class 0 probability

The probability that all N requests are to a certain group of k memories, such that none of these k memories is M_h , is p_0^N . The number of ways N requests can be distributed among k memory modules in the group, such that none of them is empty, is given by $k!S(N, k)$, where $S(N, k)$ is the Stirling number of the second type [25]. This group of k memory modules can be chosen from $(M - 1)$ memory modules in $\binom{M-1}{k}$ ways. Therefore, $\text{Pr}(E_k^{(0,N)})$ is given by

$$\text{Pr}(E_k^{(0,N)}) = k!S(N, k) \binom{M-1}{k} p_0^N. \quad (3)$$

3.2 Determination of Class 1 probability

The number of ways of choosing k modules out of M modules is $\binom{M}{k}$. Number of ways of choosing k modules out of M modules such that M_h is not included is given by $\binom{M-1}{k}$. Therefore, the number of ways in which k modules can be selected out of M modules, such that M_h is always included, is $\binom{M}{k} - \binom{M-1}{k}$.

In Class 1 type of reference, k memory modules are requested such that M_h is always requested. Therefore, out of the N references made per cycle, there could be from 1 to

$N - (k - 1)$ references to M_h and the rest will of course be to the other $(k - 1)$ non-hot memory modules M_j s, $j \neq h$. Let i be the number of references to M_h . Probability of the event $E_k^{(i, N-i)}$ is therefore given by

$$\Pr(E_k^{(i, N-i)}) = p_h^i p_0^{N-i}. \tag{4}$$

These i references to M_h can be selected from N references in $\binom{N}{i}$ ways. The number of ways in which the remaining $(N - i)$ references can be distributed among the $(k - 1)$ non-hot modules is given by

$$(k - 1)!S(N - i, k - 1).$$

Therefore, the probability that a certain group of k memory modules will be referenced, such that M_h is always included, is given by

$$\sum_{i=1}^{N-k+1} (k - 1)!S(N - i, k - 1) \binom{N}{i} p_h^i p_0^{N-i}. \tag{5}$$

As mentioned previously, a group of k memory modules can be selected out of M modules, such that M_h is always included, in $\binom{M}{k} - \binom{M-1}{k}$ ways. The probability that *any group* of k memory modules will be referenced, such that M_h is always included, is given by

$$\Pr(E_k^{(i, N-i)}) = \left(\binom{M}{k} - \binom{M-1}{k} \right) \sum_{i=1}^{N-k+1} (k - 1)!S(N - i, k - 1) \binom{N}{i} p_h^i p_0^{N-i}, \quad i \neq 0. \tag{6}$$

3.3 Average memory bandwidth

From Equations (1) and (2), the average memory bandwidth can be expressed as

$$BW = \sum_{k=1}^M k \left[\Pr(E_k^{(0, N)}) + \Pr(E_k^{(i, N-i)}) \right], \quad i \neq 0. \tag{7}$$

Substituting Equations (3) and (6) in (7) gives

$$BW = \sum_{k=1}^M k \left[k!S(N, k) \binom{M-1}{k} p_0^N + (k - 1)! \left(\binom{M}{k} - \binom{M-1}{k} \right) \sum_{i=1}^{N-k+1} S(N - i, k - 1) \binom{N}{i} p_h^i p_0^{N-i} \right]. \tag{8}$$

Table 1
Comparison of simulation results with closed-form results for $N = 20, M = 20$

p_h	BW (closed form)	BW (simulation)	Percentage error
0.1	12.6797	12.7188	-0.31
0.2	11.9511	11.9710	-0.16
0.3	11.0310	11.0482	-0.15
0.4	9.991	10.0492	-0.5
0.5	8.8540	8.8537	0.003
0.6	7.5851	7.5615	0.31
0.7	6.1798	6.1505	0.47
0.8	4.6241	4.5982	0.56
0.9	2.9031	2.8722	1.06

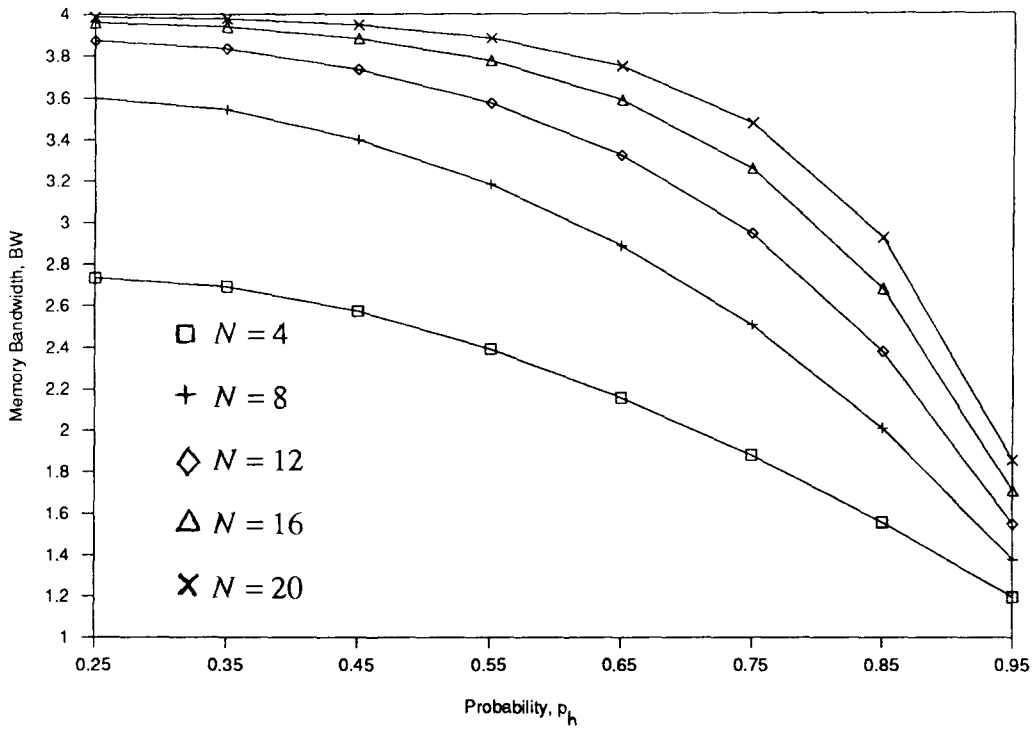


Fig. 2. Memory bandwidth vs. probability of hot memory reference for $M = 4$.

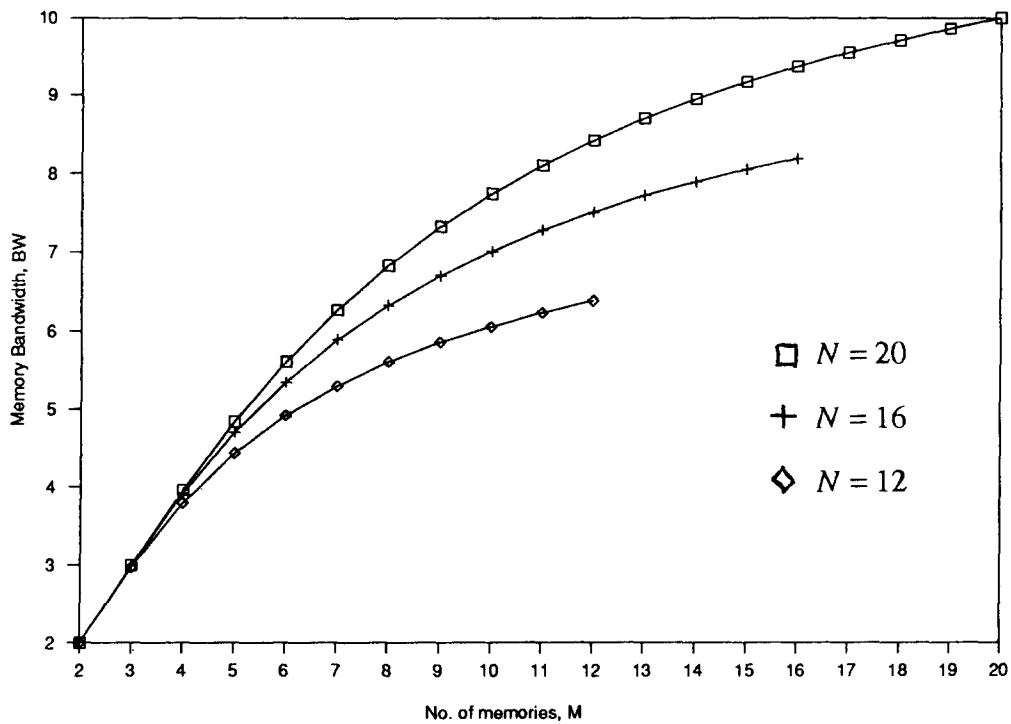


Fig. 3. Memory bandwidth vs. number of memory modules for $p_h = 0.4$.

4. Simulation results

Simulations were carried out to test the correctness of the analytical expressions developed in Section 3. The simulator was written based on the assumptions in Section 2. $N = 20$ and $M = 20$ were used. The simulator was driven by memory requests generated from a non-uniform random number generator having a distribution as described in Section 2. The simulation was performed for 4000 memory cycles. The average bandwidth was determined by observing the number of active memory modules in each cycle, and averaging them over the total number of cycles. *Table 1* shows the average memory bandwidths obtained using the closed-form solution and the simulation. Percentage errors between the two methods are also given.

Figure 2 shows the average memory bandwidth (computed using Equation 8) vs. probability of reference to a hot module, for different number of processors. As expected, the greater the number of processors the higher is the bandwidth. Note that $p_h = 0.25$ corresponds to the special case of uniform reference model. *Figure 3* shows the increase in memory bandwidth as a function of the number of memory modules, for different values of N and p_h .

5. Conclusion

Hot spots arising in multiprocessor systems due to synchronization primitives, seriously degrade the performance of crossbar multiprocessor systems. Based on probabilistic approaches, an analytical expression for calculating the average memory bandwidth of crossbar systems having a single hot spot has been developed. Simulation results have been found to be in close agreement with the analytical results. Our results can also be used in determining the bandwidth of interleaved memories in single-processor systems (with look-ahead capability), where a certain memory module has a higher probability of reference.

References

- [1] T.Y. Feng, A survey of interconnection networks, *Computer* (Dec 1981) 12–27.
- [2] T.N. Mudge, J.P. Hayes and D.C. Winsor, Multiple-bus architectures, *Computer* (Jun 1987) 42–48.
- [3] Q. Yang and L.N. Bhuyan, Performance of multiple-bus interconnections for multiprocessors, *J. Parallel Distributed Comput.* 8 (1990) 267–273.
- [4] L.N. Bhuyan, Q. Yang and D.P. Agrawal, Performance of multiprocessor interconnection networks, *Computer* (Feb. 1989) 25–37.
- [5] Y.C. Liu and C.J. Jou, Effective memory bandwidth and processor blocking probability in multiple-bus systems, *IEEE Trans. Comput.* C-36 (6) (Jun. 1987) 761–764.
- [6] T. Lang, M. Valero and I. Alegre, Bandwidth of crossbar and multiple-bus connections for multiprocessors, *IEEE Trans. Comput.* C-31 (12) (Dec 1982) 1227–1234.
- [7] P.J. Willis, Derivation and comparison of multiprocessor contention measures, *IEE Proc.* Pt. E, 1 (3) (Aug. 1978) 93–98.
- [8] L.N. Bhuyan, A combinatorial analysis of multibus multiprocessors, *Internat. Conf. Parallel Processing* (Aug. 1984) 225–227.
- [9] Q. Yang and L.N. Bhuyan, Analysis of packet-switched multiple-bus multiprocessors, *IEEE Trans. Comput.* 40 (3) (Mar. 1991) 352–357.
- [10] T.N. Mudge and H.B. Al-Sadoun, A semi-markov model for the performance of multiple-bus systems, *IEEE Trans. Comput.* C-34 (10) (Oct. 1985) 934–942.
- [11] D.W.L. Yen, J.H. Patel and E.S. Davidson, Memory interference in synchronous multiprocessor systems, *IEEE Trans. Comput.* C-31 (11) (Nov. 1982) 1116–1121.
- [12] T.N. Mudge, J.P. Hayes, G.D. Buzzard and D.C. Winsor, Analysis of multiple-bus interconnection networks, *J. Parallel Distributed Comput.* 3 (1986) 328–343.

- [13] M.A. Marsan and M. Gerla, Markov models for multiprocessor systems, *IEEE Trans. Comput.* C-31 (3) (Mar. 1982) 239–248.
- [14] A Fukuda, Equilibrium point analysis of memory interference in multkprocessor systems, *IEEE Trans. Comput.* 37 (5) (May 1988) 585–593.
- [15] D. Towsley, Approximate models of multiple bus multiprocessor systems, *IEEE Trans. Comput.* C-35 (3) (Mar. 1986) 220–228.
- [16] C.R. Das and L.N. Bhuyan, Bandwidth availability of multiple-bus multiprocessors, *IEEE Trans. Comput.* C-34 (10) (Oct 1985) 918–926.
- [17] L.N. Bhuyan, An analysis of processor-memory interconnection networks, *IEEE Trans. Comput.* C-34 (3) (Mar 1985) 279–283.
- [18] A.S. Sethi and N. Deo, Interference in multiprocessor systems with localized memory access probabilities, *IEEE Trans. Comput.* C-28 (2) (Feb. 1979) 157–163.
- [19] K.B. Irani and I.H. Onyuksel, A closed-form solution for the performance analysis of multiple-bus multiprocessor systems, *IEEE Trans. Comput.* C-33, (11) (Nov. 1984) 1004–1012.
- [20] K.O. Siomalas and B.A. Bowen, Performance of cross-bar multiprocessor systems, *IEEE Trans. Comput.* C-32 (7) (Jul. 1983) 689–695.
- [21] T.N. Mudge and B.A. Makrucki, Probabilistic analysis of a crossbar switch, *9th Annual Symp. on Computer Architecture* (Apr. 1982) 311–210.
- [22] G.F. Pfister and V.A. Norton, Hot spot contention and combining in multistage interconnection networks, *IEEE Trans. Comput.* C-33 (10) (Oct. 1985) 943–948.
- [23] G. Lee, C.P. Kruskal and D.J. Kuck, The effectiveness of combining in shared memory parallel computers in the presence of hot spots, *1986 Internat. Conf. Parallel Processing* (Aug. 1986) 35–41.
- [24] S.L. Scott and G.S. Sohi, The use of feedback in multiprocessors and its applications to tree saturation control, *IEEE Trans. Parallel. Distributed Syst.* 1 (4) (Oct. 1990) 385–398.
- [25] J. Bradley, *Introduction to Discrete Mathematics* (Addison-Wesley, Reading, MA, 1988).