Computation Availability of Crossbar 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

Multiprocessor systems can provide higher performance, reliability, and availability than single processor systems. In order to properly assess the effectiveness of multiprocessor systems, measures that combine performance and reliability are needed. Most previous studies have considered the performance and reliability/availability separately. Only a limited number of studies have been carried out by considering the performance and reliability together in the presence of a uniform traffic in the network. Hot spots in a multiprocessor system result in a non uniform traffic in the network and results in a degradation in the performance of the network. The aim of this paper is to study the combined performance and reliability of a crossbar system in the presence of a hot spot in the system. Computation availability is taken as the criteria for measuring the combined performance and reliability. Markov chain models have been developed to compute the above measurement criteria.

1 Introduction

Multiprocessor systems can provide higher performance, reliability, and availability than single processor systems. Failure in the processor of a single processor system may render the system unusable. On the other hand, a faulty processor in a multiprocessor system can be isolated from the system with the system being still able to function at a reduced level of performance. Most of the previous studies have considered the performance and reliability/availability of multiprocessor systems separately. The performance of a multiprocessor system is affected by failures in the system. In a gracefully degradable system, the system resources can be reconfigured by isolating the faulty component to carry out useful work at a reduced level of performance. Therefore, the simultaneous consideration of the issues of performance and fault tolerance is very important. Measures that combine performance and reliability are needed to properly assess the effectiveness of multiprocessor systems.

This paper aims to address two fault-tolerant issues, namely, computation availability and bandwidth. Reliability of a non-repairable system at time t is defined to be the probability that the system is operational during the interval $(0, t)$, given that it was operational at $t = 0$. Computation availability [1] at time t is defined as the amount of computation available at time t.

The computation availability is based on the system requirements for the concurrent execution of a task that consists of a few subtasks. Hence, it is also sometimes referred to as the *task-based computation availability*, which is defined as the probability that the system has at least the minimum number of resources working at time t for the execution of a task, and that there exists a valid communication path between the resources. In a degradable multiprocessor system, the minimum number of resources must be available for the execution of the subtasks of a task. Task-based computation availability measure represents the performance of the system (to execute the task) at different instants of time.

We will assume a crossbar system for our study. A crossbar system does not suffer from bus conflicts and provides a high bandwidth. A crossbar system is suitable for small to medium sized systems. Performance results of crossbar systems in the case of uniform memory reference pattern has been reported in [2, 3]. Favorite memory case has been analyzed in [4]. However, the assumption of uniform memory reference pattern in a multiprocessor system is not realistic. Different non-uniform traffic patterns have been reported in the literature [5, 6, 7, 8]. Hot spot traffic is a particular type of non-uniform traffic which arises due to the use of synchronization primitives and shared variables in a shared-memory system. The effect of hot spots have been found to degrade the performance of multiprocessor systems using crossbar [9], multiple bus [10], and multistage interconnection networks [11, 12].

Reliability studies of different networks can be found in [13, 14, 15, 16, 17]. Since performance or reliability separately is not sufficient to properly describe the effectiveness of a system, reliability based performance measures have been used to measure the usefulness of a fault tolerant system. Most of the previous work on performance based reliability have assumed a uniform traffic pattern in the network. *The objective of this paper is to study the performance based reliability of a crossbar multiprocessor system in the presence of a non uniform (hot spot) traffic in the network.* We assume a fault tolerant crossbar multiprocessor system, thereby implying a gracefully degradable system. We have used a task based reliability model for out study of performance. A task based model assumes that a minimum number of system resources are needed for the completion of a task. *Computation availability* (CA) has been assumed as a measure of the effectiveness of the system. CA at any time t is a measure of the amount of computation that is available at time t. It is expressed as the product of the bandwidth and the probability that the system is functioning at time t. In this paper, we develop models for the performance and reliability to arrive at a model for computation availability.

The rest of this paper is organized as follows. The assumptions required for the modeling are described in Section 2. The bandwidth of a crossbar system in the presence of a non uniform traffic pattern is described in Section 3. Section 4 deals with a model of the computation availability. Results obtained from the model are presented in Section 5. The last section summarizes the results.

2 Modeling Assumptions

The following assumptions are made regarding the operation of the network.

1. The system consists of N identical processors and M identical memories. The processors and memory modules will be represented by $\{\mathrm{PE}_0, \mathrm{PE}_1, \cdots, \mathrm{PE}_{N-1}\}$ and $\{\mathrm{MM}_0, \mathrm{MM}_1, \cdots, \mathrm{MM}_{M-1}\}$ respectively.

2. The processors and memories are connected by a *crossbar* interconnection network. The interconnection can be thought of as having M busses, where each memory module is connected to only a bus and each processor is connected to all the busses (see Figure 1).

3. The system operates *synchronously.* Each processor generates a request at the start of a memory cycle with probability r. All memory modules having requests at the beginning of a memory cycle provide the requested data at the end of the cycle.

4. Requests from the processors during a cycle are assumed to be *spatially independent*, i.e. requests from the different processors during a cycle are independent of each other.

5. Requests from a processor at successive cycles are assumed to have *temporal independence.* This implies that requests which can not be accepted during a cycle are simply rejected.

6. Resolution of memory conflicts by the bus arbiter is unbiased, i.e., the conflicts are resolved *randomly* by the bus arbiter.

7. MM_h is assumed to be a hot memory module for all processors. The probability that $\mathrm{PE}_i, 0 \leq i \leq N-1$, requests MM_h is p_h, and that it requests $\mathrm{MM}_j, j \neq h$, is $p_0 = (1 - p_h)/(M-1)$, where $p_h > 1/M$. The *factor of hotness* is denoted by h, and it represents the factor by which the hot memory is likely te be referenced as compared to a non hot memory. Therefore, $p_h = \frac{h}{M+(h-1)}$. Note that $h = 1$ is the uniform memory reference model that is widely used in the literature.

8. The failures of a processor, a memory, and a bus are assumed to be exponentially distributed with rates λ_p, λ_m, and λ_b respectively. It is also assumed that other components of the system (for example, the bus arbiter) do not fail.

9. The coverage factor [18] of a state of the system is the probability that the system reconfigures correctly, given that it has detected a failure in a module. We assume a coverage factor of unity.

Though assumption 5 is unrealistic, it makes the modeling simple without introducing too much of errors [19].

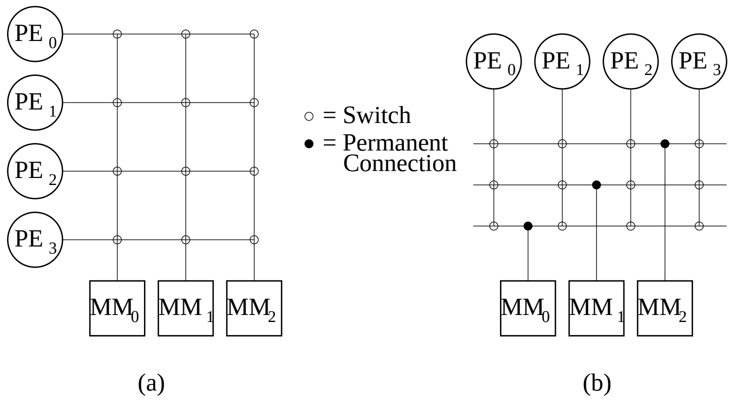

Figure 1: (a) A 4×3 crossbar system, (b) and its representation in the form of a bus

3 Determination of Bandwidth

The average memory bandwidth of an $N \times M$ crossbar system is defined as the expected number of memory requests that are accepted by the memory modules during a cycle. It is expressed as

$$\mathrm{BW}_{N,M} = \sum_{k=1}^{M} k \Pr(E_k) \tag{1}$$

where, $\Pr(E_k)$ is the probability of the event that k memory modules are requested during a cycle. Let $E_k^{p,q}$ be the event that k distinct memory modules are requested during a cycle, and p and q are the number of requests for the hot memory module and the set of non-hot memory modules respectively, $0 \leq p+q \leq N$. Let $n = p+q$ be the total number of requests generated during a cycle. Therefore, the probability that k distinct memories are requested during a cycle is given by

$$\begin{aligned} \Pr(E_k) &= \Pr(E_k^{i,n-i}), \qquad 0 \leq i \leq n \\ &= \Pr(E_k^{0,n}) + \Pr(E_k^{i,n-i}), \quad 1 \leq i \leq n \end{aligned} \tag{2}$$

Substituting equation 2 in 1 gives

$$\mathrm{BW} = \sum_{k=1}^{M} k(\Pr(E_k^{0,n}) + \Pr(E_k^{i,n-i})) \quad 1 \leq i \leq n \tag{3}$$

where, the event $E_k^{0,n}$ and $E_k^{i,n-i}, i \neq 0$, are defined as class 0 and class 1 events respectively. The probability of having exactly n requests from N processors at the beginning of a memory cycle is $\binom{N}{n} r^n (1-r)^{N-n}$. We calculate the class 0 and class 1 events as follows.

3.1 Calculating the Class 0 Probability

Given that n requests have been generated during a cycle, the probability that all the requests are directed to k distinct non-hot memory modules can be given by [9]

$$\Pr(E_k^{0,n} | n \text{ requests generated}) = k! S(n,k) \binom{M-1}{k} p_0^n \tag{4}$$

where, $S(n,k)$ is the Stirling number [20] of the second type. Therefore, class 0 probability is given by [21]

$$\Pr(E_k^{0,n}) = \sum_{n=1}^{N} k! S(n,k) \binom{M-1}{k} p_0^n \binom{N}{n} r^n (1-r)^{N-n} \tag{5}$$

3.2 Calculating the Class 1 Probability

The calculation of class 1 probability is carried out using the same approach as used in the determination of the class 0 probability. The number of ways in which k memories can be selected out of M memories such that MM_h is always included is given by $\binom{M}{k} - \binom{M-1}{k}$. The probability of i references being to hot memory and $(n-i)$ to the non hot memories is $p_h p_0^{n-i}$. These i references to MM_h can be selected from n references in $\binom{n}{i}$ ways, and the remaining $(n-i)$ references can be distributed among the $(k-1)$ non hot memory modules in $(k-1)!S(n-i,k-1)$ ways. Therefore, the probability that a certain set of k memory modules are requested such that MM_h is one of the modules in the set is given by

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

The probability of the class 1 event is therefore given by

$$\begin{aligned}\Pr(E_k^1) = \sum_{n=1}^{N}\left(\binom{M}{k} - \binom{M-1}{k}\right) \\ \times \sum_{i=1}^{n-k+1} (k-1)!S(n-i,k-1)\binom{n}{i}p_h^i p_0^{n-i}\binom{N}{n}r^n(1-r)^{N-n}, i \neq 0\end{aligned} \tag{6}$$

3.3 Average Memory Bandwidth

Average memory bandwidth is obtained by substituting equations (5) and (6) in (3) It is given by

$$\begin{aligned}\mathrm{BW}_{\mathrm{N,M}} = \sum_{k=1}^{M} k \left\{ \sum_{n=1}^{N} k!S(n,k)\binom{M-1}{k}p_0^n\binom{N}{n}r^n(1-r)^{N-n} + \sum_{n=1}^{N}\left(\binom{M}{k} - \binom{M-1}{k}\right)\right. \\ \left. \times \sum_{i=1}^{n-k+1} (k-1)!S(n-i,k-1)\binom{n}{i}p_h^i p_0^{n-i}\binom{N}{n}r^n(1-r)^{N-n}\right\}\end{aligned} \tag{7}$$

4 Computation Availability Modeling

Markov modeling will be used to determine the CA for a task requiring at least I processors and J memories [4]. The Markov chain is similar to that in [4]. Let $P_{i,j}(t)$ denote the probability of availability of the system at time t, where i and j are the number of processors and memories operational at time t, given that all the components of the system were operational at $t = 0$. The CA is therefore, given by

$$\mathrm{CA}(t) = \sum_{i=I}^{N}\sum_{j=J}^{M} P_{i,j}(t)\mathrm{BW}_{i,j} \tag{8}$$

where, $\mathrm{BW}_{i,j}$ is the bandwidth of an $i \times j$ crossbar system obtained by replacing N and M by i and j respectively in equation 7. $P_{i,j}(t)$ is given by the product of the reliabilities of the processors, memories, and busses.

$$P_{i,j}(t) = \binom{N}{i}\left(R_p(t)\right)^i\left(1-R_p(t)\right)^{N-i} \times \binom{M}{j}\left(R_e(t)\right)^j\left(1-R_e(t)\right)^{M-j} \tag{9}$$

where, $R_p(t) = e^{-\lambda_p t}$ is the reliability of a processor and $R_e(t) = e^{-(\lambda_m+\lambda_b)t}$ is the equivalent reliability of a memory-bus combination at time t. Substituting equations (5) and (7) in (8) gives the required computation availability at time t.

5 Results

The variation in computation availability with respect to time for $r = 1$ and tasks requiring different number of processors and memories in a 16×16 crossbar system is shown in Figure 2. $\lambda_p = 0.0001$, $\lambda_m = 0.00005$, and $\lambda_b = 0.0001$, have been used throughout the paper. The curves with the solid and dotted lines represent the computation availability for uniform traffic ($h = 1$) and hot spot traffic ($h = 4$)

respectively. It is found that the CAs for the task requiring all the processors ($I = J = 16$) are very close for $h = 1$ and $h = 4$. This is because the task can not tolerate even a single fault, and hence the CA is dominated by the reliability of the system. On the other hand, for a task requiring 8 processors and 8 memories, the CA for $h = 4$ is significantly lower than that for $h = 1$. This is due to the fact that the task can tolerate a large number of faults, and hence the CA is dominated by the bandwidth, which is significantly higher for uniform traffic ($h = 1$) as compared to hot spot traffic ($h = 4$).

The computation availability for a system having a network input rate of $r = 0.5$ is shown in Figure 3. The variation of CA follows a similar pattern as in Figure 2. The values of the CAs are, of course, much lower than for $r = 1$. This is due to the lower bandwidth for $r = 0.5$.

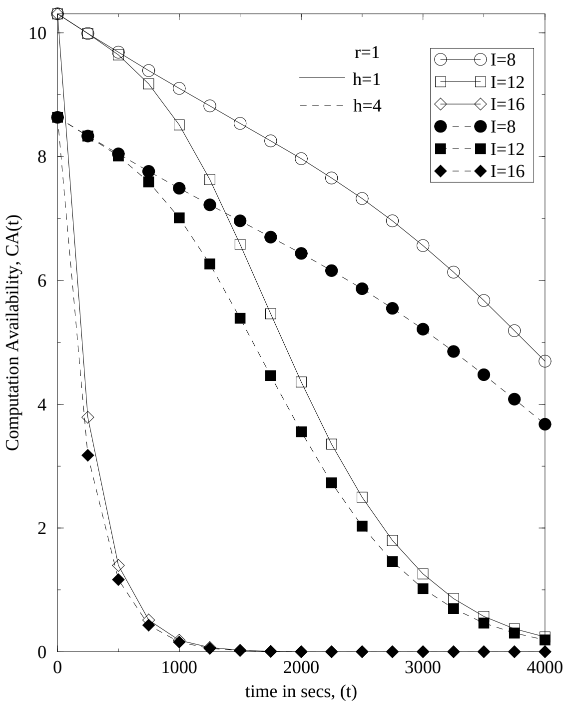

Figure 2: Variation of computation availability for tasks requiring different amounts of resources for r=1.0

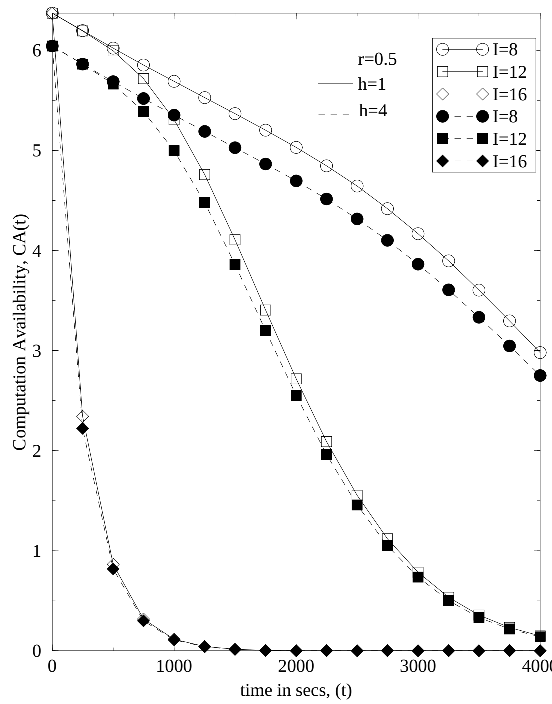

Figure 3: Variation of computation availability for tasks requiring different amounts of resources for r=0.5

6 Conclusions

Task-based computation availability provides a better measure of the effectiveness of a multiprocessor system for executing a task. The task consists of several sub-tasks which require a minimum number of resources for their execution. A model for task based computation availability has been developed by assuming graceful degradation of the system. Our model differs from the previous models in the sense that we have assumed a hot spot traffic pattern in the network, whereas previous models have assumed uniform traffic in the network.

Computation availability for tasks requiring varying number of system resources under different network input traffic loads have been presented. Computation availability in the presence of a hot spot traffic has been compared with that in the presence of uniform traffic in the network.

The model can be extended to take care of a coverage factor which is less than unity. Extension to the model is currently being carried out.

References

[1] M.D. Beaudry, "Performance related reliability measures for computing systems," *FTCS-7: 7th Annual International Conference on Fault Tolerant Computing*, pp. 16–21, June 28-30, 1977.

[2] T. Lang, M. Valero, and I. Alegre, “Bandwidth of crossbar and multiple-bus connections for multiprocessors,” *IEEE Transactions on Computers*, vol. C-31, no. 12, pp. 1227–1234, December 1982.

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

[4] C.R. Das and L.N. Bhuyan, “Bandwidth availability of multiple-bus multiprocessors,” *IEEE Transactions on Computers*, vol. C-34, no. 10, pp. 918–926, October 1985.

[5] S.L. Scott and G.S. Sohi, “The use of feedback in multiprocessors and its applications to tree saturation control,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, no. 4, pp. 385–398, October 1990.

[6] J.S.C. Chen and T.E. Stern, “Throughput reduction due to non-uniform traffic in a packet switch with input and output queueing,” *International Conference on Communications*, Denver, Colorado, pp. 413–417, June 23-26, 1991.

[7] S.Abraham and K. Padmanabhan, “Performance of direct binary n-cube network for multiprocessors,” *IEEE Transactions on Computers*, vol. 38, no. 7, pp. 1000–1011, July 1989.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] J.T. Blake, “Reliability analysis of interconnection networks using hierarchical composition,” *IEEE Transactions on Reliability*, vol. 38, no. 1, pp. 111–120, April 1989.

[14] J. Kim, C.R. Das, W. Lin, and T.Y. Feng, “Reliability evaluation of hypercube multicomputers,” *IEEE Transactions on Reliability*, vol. 38, no. 1, pp. 121–129, April 1989.

[15] A. Varma and C.S. Raghavendra, “Reliability analysis of redundant-path interconnection networks,” *IEEE Transactions on Reliability*, vol. 38, no. 1, pp. 130–137, April 1989.

[16] C. Botting, S. Rai, and D.P. Agrawal, “Reliability computation of multistage interconnection networks,” *IEEE Transactions on Reliability*, vol. 38, no. 1, pp. 138–145, April 1989.

[17] A.D. Ingle and D.P. Siewiorek, “Reliability models for multiprocessor systems with and without periodic maintenance,” *FTCS-7: 7th Annual International Conference on Fault Tolerant Computing*, pp. 3–9, June 28-30, 1977.

[18] T.F. Arnold, “The concept of coverage and its effect on the reliability model of a repairable system,” *IEEE Transactions on Computers*, vol. C-22, no. 3, pp. 251–254, March 1973.

[19] J.H. Patel, “Performance of processor-memory interconnections for multiprocessors,” *IEEE Transactions on Computers*, vol. C-30, no. 10, pp. 771–780, October 1981.

[20] J. Bradley, *Introduction to Discrete Mathematics.* Addison-Wesley Pub;ishing Company, 1988.

[21] M. A. Sayeed and M. Atiquzzaman, “Performance of multiple-bus multiprocessor under non-uniform memory reference model,” *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, North Carolina, USA, January 31 - February 2, 1994.