Memory Interference Models with Variable Connection Time

T. N. MUDGE and HUMOUD B. AL-SADOUN

Abstract — This correspondence develops two discrete memory interference models. These models, the equivalent rate model and the Markov chain model, allow for variable connection times between processors and memories if these times can be characterized by a discrete random variable \( X \). The equivalent rate model, which is the simpler, requires only the first two moments of \( X \), while the Markov chain model requires the first and second moments. The models yield estimates of the bandwidth \( BW \), the probability of acceptance \( P_a \), and processor utilization \( U_p \). Both models give good estimates of \( BW \) when the coefficient of variation \( C_x \) of \( X \) is small. When \( C_x \) reaches 2.0 the Markov chain model still shows an error of less than 4 percent while the equivalent rate model exhibits a 50 percent error that, unlike the Markov chain model, continues to increase with increase in \( C_x \). Finally, it is shown that \( BW \) drops significantly with increase in \( C_x \), suggesting that processor–memory transfers should use a fixed block size if memory conflict is to be minimized.

Index Terms — Markov chains, memory bandwidth, memory interference, multiprocessors, performance evaluation.

I. INTRODUCTION

This correspondence develops two discrete time models of the memory interference that occurs during memory access in a multiprocessor system. These models, termed the equivalent rate (ER) model and the Markov chain (MC) model, provide for variable connection times between processors and memory modules if these connection times can be characterized by a discrete random variable \( X \). The models assume a synchronous multiprocessor having \( N \) processors and \( M \) memory modules. The processors share the memory modules through an \( N \times M \) crossbar interconnection network. The whole system is synchronized with a system clock whose period is referred to as “the system cycle.” The discrete values taken on by \( X \) are in units of system cycles. A processor–memory transfer typically involves an initial cycle to allow for decoding the address of the memory module and resolving any interference followed by subsequent (\( \geq 1 \)) data transfer cycles. To minimize the loss in useful accessing time due to the “setup” cycle, the average connection time should be made as large as possible.

The literature contains a number of memory interference models for multiprocessor systems (see [1]–[11]). In these studies system operation is approximated by a stochastic process as follows. At the beginning of the system cycle a processor selects a memory module at random and makes a request to access that module with probability \( r (\leq 1) \). If more than one request is made to the same memory module, it will choose one at random; the other processors will retry in the next cycle. A processor has at most one pending request waiting for access at any time. The behavior of the processors is considered to be independent but statistically identical. A processor that obtains a connection to a memory module at the beginning of the system cycle will release that module at the end of the cycle.

A model for the system described above results in a Markov chain having an unmanageably large state space (see [1] and [4]). One of the main themes of the work in [1]–[11] is to develop models which avoid this complexity while maintaining reasonable agreement with simulation results. This is done by further simplifying the assumptions of the system behavior. The models develop equations for the memory bandwidth \( BW \), the probability of acceptance \( P_a \), and in some cases processor utilization \( U_p \). In [10] a classification has been proposed for these models according to the approach used in their formulation. The classes are: probabilistic models typified by the models in [2], [3], [7]–[9], [11]; and [12]; rate-adj usted probabilistic models typified by the models in [5] and [6]; queueing system models typified by the models in [4] and [7]; and steady state flow models typified by the model in [10].

In addition to the above discrete time models, continuous time models have been proposed. These are typified by the models in [13] and [14], in which the memory–processor connection time is denoted by an exponentially distributed random variable and the processor interrequest time is also denoted by an exponentially distributed random variable. Therefore, unlike the discrete time models, these models can accommodate variable connection times provided the times can be approximated by an exponentially distributed random variable. However, continuous time models are usually less accurate than discrete time models when discrete time events are being modeled [16], [17].

II. THE SYSTEM OPERATION ASSUMPTIONS

A processor can be in any of three states: thinking, when it has no outstanding request to memory (it might be performing local processing); accessing, when it is connected to a memory module; and waiting, when it has a pending memory request waiting to be serviced. The memory module can be in one of two states: busy, when it is being accessed by a processor; and idle, when it is not being accessed. The following assumptions further characterize the operation of the system.

1) Processors’ requests for memory form independent statistically identical stochastic processes.

2) At the beginning of a system cycle a processor in the thinking state or one that has just completed a memory access makes a request to access a memory module with probability \( r \).

3) If more than one processor issues a request to a particular memory module, that memory, if idle, will choose a processor at random to get the connection. The other processors will retry to the same memory module in the next cycle. Requests to busy memories will also retry to the same memory in the next cycle.

4) The requests originating from the same processor are independent of each other. In each case a memory module will be chosen at random from the \( M \) memory modules with equal probability, i.e., with probability \( 1/M \).

These models also considered the case, not considered here, of multiple-bus interconnections rather than a crossbar. Until recently no simple discrete time model existed for this [15].
5) The connection time, in units of system cycles, between a processor and a memory module is determined by a discrete random variable $X$, which has a probability mass function $f(i)$, i.e., $f(k) = \Pr[X = k]$.

Assumptions 1)–4) are common to all the discrete models mentioned earlier. Empirical evidence to support their adoption can be found in [3]–[5] among others. Assumption 5) is the key difference between the models mentioned earlier and those developed here; thus, the models developed in this paper depend on the extent to which connection times can be approximated by a random variable $X$ having a probability mass function $f(i)$.

In order to derive numerical information from the memory interference models developed later, the values of $M, N, r, \text{and} r$ and the first two moments of $X$ must be obtained through measurement or, if it is considered satisfactory, by hypothesis. These quantities can be regarded as the inputs to the models.

The performance measures that will be derived from our models are: memory bandwidth $\text{BW}$, the probability of acceptance $P_a$, and processor utilization $U_p$. These quantities can be regarded as the outputs of the models. The memory bandwidth is defined as the expected number of busy memory modules seen by a random arrival after the system reaches its steady state or, equivalently, the expected number of accessing processors seen by a random arrival after the system reaches its steady state. The probability of acceptance is defined as the probability that a memory request is accepted. Finally, processor utilization is defined as the fraction of time a processor spends thinking or accessing a memory after the system has reached steady state.

III. MEMORY INTERFERENCE MODELS

A. Equivalent Rate Model (ER Model)

This model is a modification of an approach first presented in [18] to study the behavior of multiprocessor systems in which each processor has a private cache memory. The processor-memory connections were assumed to be a fixed number of system cycles—the time needed to transfer one cache line. A probabilistic model (see [10]) was used to compute $\text{BW}$, and an equivalent value for $r$ (the equivalent rate) was derived from the fixed number of system cycles needed to transfer a line and the probability of requesting a line transfer. In our ER model the foregoing approach is retained; the main differences are, first, that the equivalent rate is modified to take into account the fact that processor-memory transfers take a variable number of system cycles and, secondly, that the probabilistic model is replaced by the steady state flow model (see [10]) because it produces a smaller error.

The derivation of the equivalent rate $r_{eq}$ proceeds as follows. The average connection time in units of system cycles can be expressed by

$$\bar{X} = \sum_{i=1}^{5} i f(i)$$

i.e., the first moment of the random variable $X$ ($5$ is the maximum connection time). From assumption 2) the thinking time is geometrically distributed. Therefore, if $\bar{T}$ is the average thinking time, it follows that

$$\bar{T} = \frac{1 - r}{r}$$

The equivalent rate can now be defined by

$$r_{eq} = \frac{\bar{X}}{\bar{X} + \bar{T}}.$$  

This is the fraction of time a processor is accessing memory assuming no interference. The $\text{BW}$ can now be obtained by solving the following two equations iteratively:

$$\text{BW} = NU_p r_{eq}$$

$$\text{BW} = M \left(1 - \left(1 - \frac{U_p r_{eq}}{M}\right)^N\right) \left(1 - \frac{1}{M} \left(1 - \left(1 - \frac{U_p}{M}\right)^N\right)ight).$$

As will be shown, the ER model captures the average $\text{BW}$ behavior of multiprocessor systems only if the coefficient of variation $^2\text{CV}_X$ is zero. It introduces inaccuracies in those cases where there is randomness in the connection time because it does not take into account the second moment of the connection time. The ER model is not appropriate for determining $P_a$ because the flow model of [10] that gives rise to the above equations implicitly has $P_a = U_p = \text{BW}/N r_{eq}$.

B. Markov Chain Model (MC Model)

The Markov chain which models a multiprocessor system according to the assumptions outlined in Section II has an unmanageably large state space (see [1] and [4]). The MC model dramatically reduces the size of this state space by making two further simplifying assumptions. The first assumes it is sufficient to describe the behavior of only one processor. Within the framework of the original assumptions this is sufficient to describe the complete system behavior because all processors are assumed independent and statistically identical [assumption 1)]. When a processor is blocked in an attempt to access a memory module after placing a request, it enters a series of waiting states for the residual service time of that module. The residual service time (RST) of a memory module is the time remaining before the currently accessing processor releases the memory module. The second assumption, similar to that used in the rate adjusted probabilistic models (see [5] and [6]), assumes that the processor waits for the RST before placing a new and independent request with a probability of 1.0. The request is directed to any memory module independent of the particular memory it was previously blocked at. This assumption, 3a), is a relaxation of assumption 3) which requires that resubmission be.write to the same memory. The above two assumptions allow simplification of the Markov chain but usually cause an overestimate of the memory bandwidth of the system being modeled.

The Markov chain, shown in Fig. 1, defines a processor’s behavior if assumption 3a) is used. When a processor is in state $i$, it is accessing a memory module and it needs $i$ more cycles before releasing the connection. After one cycle in state $i$ the processor always moves to state $i - 1$, indicating it needs $i - 1$ more cycles. Thus, there is a single transition from state $i$ ($1 < i \leq S$) to state $i - 1$ with a probability of 1.0. The set of states $[i]$ are accessing states. When a processor is in state 0 it is in the thinking state and it may be performing local processing. When a processor is in state $i$ it had a memory request blocked and must wait $i$ cycles before it can resubmit the request. After one cycle in state $i$ the processor always moves to state $i - 1$, indicating it must wait $i - 1$ more cycles. Thus, there is a single transition from state $i$ ($1 < i \leq S$) to

$$^2\text{CV}_X = \frac{\text{standard deviation of } X}{\text{expected value of } X} = \sqrt{(X^2)/E(X) - 1}$$

where $X$ and $X^2$ are the first and second moments of $f(i)$, respectively.
state \( i \rightarrow T \) with a probability of 1.0. The set of states \( \{i\} \) are waiting states. The transition probabilities \( \alpha_i, \beta_i, \overline{\alpha}_i, \) and \( \overline{\beta}_i \) are defined by the following equations:

\[
\alpha_i = \begin{cases} 
1 - r & (\text{Pr}[\text{a request is made and accepted and needs } i \text{ cycles}]) \\
0 & (1 \leq i \leq S)
\end{cases}
\]

\[
\beta_i = \text{Pr}[\text{a request is made to a memory that has an RST of } i \text{ cycles}] \quad 1 \leq i \leq S
\]

\[
\overline{\alpha}_i = \text{Pr}[\text{a request is accepted and needs } i \text{ cycles}] = \frac{\alpha_i}{r} \quad 1 \leq i \leq S
\]

\[
\overline{\beta}_i = \text{Pr}[\text{a memory has an RST of } i \text{ cycles | a request was made to it}] = \frac{\beta_i}{r} \quad 1 \leq i \leq S.
\]

If the processor is in the thinking state, i.e., state 0, one of three possibilities can occur at the beginning of the next system cycle: the processor continues in the thinking state with probability \( \alpha_0 (= 1 - r) \); the processor accesses a memory module for the next \( i \) cycles with probability \( \alpha_i \), i.e., the processor enters state \( i \); or the processor is blocked for the next \( i \) cycles with probability \( \beta_i \), i.e.,

\[
(1 - P_{\text{un}})f(i) = \text{Pr}[\text{a memory request is blocked and the proc.}]
\]

\[
1 - B' = \text{Pr}[\text{a processor requests a memory that is idle}]
\]

\[
P_{i+1} \overline{B} = \text{Pr}[\text{a busy memory seen by a proc. has an RST of } i \text{ cycles}]
\]

When the processor enters state \( i \). The same three possibilities can occur if the processor is about to end its connection period at the beginning of the next system cycle, i.e., when the processor is in state 1. On the other hand, if the processor is at the end of its waiting period, i.e., it is in state \( T \), and since assumption \( 3a \) requires the re-submission of blocked requests with a probability of one, only one of these possibilities can occur at the beginning of the next system cycle: the processor gains access to a memory module for the next \( i \) cycles with probability \( \overline{\alpha}_i \), i.e., the processor enters state \( i \); or the processor is again blocked and must wait for the next \( i \) cycles with probability \( \overline{\beta}_i \), i.e., the processor enters state \( i \). There is no transition from state \( T \) to the thinking state, i.e., state 0, because of assumption \( 3a \).

Let \( P_i \) be the limiting probability for state \( i \), then the following definitions will be useful in the remainder of this paper:

\[
R \triangleq \text{Pr}[\text{a request is made at the beginning of any system cycle}]
\]

\[
= r(P_1 + P_0) + P_T
\]

\[
P_{\text{un}} \triangleq \text{Pr}[\text{a request is accepted by an idle memory at the beginning of a system cycle}]
\]

\[
= \frac{M}{NR} [1 - (1 - R/M)^N]
\]

\[
B \triangleq \text{Pr}[\text{a processor is still accessing at the end of a system cycle}]
\]

\[
= \sum_{i=2}^{N} P_i
\]

\[
B_k \triangleq \text{Pr}[\text{processor } k \text{ requests a memory that is busy}]
\]

\[
= \sum_{j=1}^{N} \text{Pr}[\text{processor } j \text{ is still accessing a memory at the end of a system cycle}]
\]

\[
= \sum_{j=1}^{N} \frac{B}{M} = \frac{(N - 1)B}{M}
\]

It follows from assumption 1) that \( B_k \) is independent of \( k \), thus we can define

\[
B' \triangleq B_k \quad 1 \leq k \leq N.
\]

One important distinction should be noted: \( R \) is the probability that a processor makes a request (including resubmissions) at the begin-

\[
(1 - P_{\text{un}})f(i) = \text{Pr}[\text{a memory request is blocked and the proc.}]
\]

\[
1 - B' = \text{Pr}[\text{a processor requests a memory that is idle}]
\]

\[
P_{i+1} \overline{B} = \text{Pr}[\text{a busy memory seen by a proc. has an RST of } i \text{ cycles}]
\]

\[
= \frac{rP_{\text{un}}(1 - B')f(i)}{1 - P_{\text{un}}}(1 - P_{\text{un}})(1 - B')f(i)
\]

\[
1 \leq i \leq S
\]

\[
\beta_i = r \left[ B' \overline{P}_{i+1} \overline{B} + (1 - B')(1 - P_{\text{un}})f(i) \right]
\]

\[
1 \leq i \leq S - 1
\]

\[
\beta_s = r(1 - B')(1 - P_{\text{un}})f(S)
\]

\[
\overline{\alpha}_i = \frac{\alpha_i}{r} \quad 1 \leq i \leq S
\]

\[
\overline{\beta}_i = \frac{\beta_i}{r} \quad 1 \leq i \leq S.
\]

The transition probabilities for the Markov chain of Fig. 1 can now be expressed as follows:

\[
\alpha_0 = 1 - r
\]

\[
\alpha_i = rP_{\text{un}}(1 - B')f(i) \quad 1 \leq i \leq S
\]

\[
\beta_i = r \left[ B' \overline{P}_{i+1} \overline{B} + (1 - B')(1 - P_{\text{un}})f(i) \right]
\]

\[
1 \leq i \leq S - 1
\]

\[
\beta_s = r(1 - B')(1 - P_{\text{un}})f(S)
\]

\[
\overline{\alpha}_i = \frac{\alpha_i}{r} \quad 1 \leq i \leq S
\]

\[
\overline{\beta}_i = \frac{\beta_i}{r} \quad 1 \leq i \leq S.
\]

The limiting probabilities for the states can be written as follows:
\[ P_i = \left( \sum_{j=1}^{s} \alpha_j \right) \frac{R}{r} \quad 1 \leq i \leq S \]
\[ P_0 = \frac{\alpha_0}{1 - \alpha_0} \left( \sum_{j=1}^{s} \alpha_j \right) \frac{R}{r} \]
\[ P_j = \left( \sum_{i=1}^{s} \beta_i \right) \frac{R}{r} \quad 1 \leq i \leq S \]

Since the above \( 2S + 1 \) equations must add to one, it follows that:
\[ R = \frac{r}{\sum_{j=1}^{s} \alpha_j + \sum_{j=1}^{s} \alpha_j (i+1)} \quad (3) \]

From the definitions of \( B' \), \( B \), and (2), \( B \) can be expressed as follows:
\[ B = \sum_{i=2}^{S} P_i = \left( \sum_{i=1}^{S} (i-1) \alpha_i \right) \frac{R}{r} \]

Substituting in the above equation for \( \alpha_0 \) from (2) allows \( B \) to be expressed as the following function of \( R \):
\[ B = \frac{(\bar{X} - 1) P_{\text{min}} R}{1 + \frac{N - 1}{M} (\bar{X} - 1) P_{\text{min}} R} \quad (4) \]

Substituting (2) into (3) results in the following equation:
\[ R = \left( 1 - \frac{N - 1}{M} B \right) \left( \bar{X} + (1/r - 1) P_{\text{min}} + \frac{(N - 1) P_{\text{min}} R}{M} \sum_{i=1}^{s} \frac{(i-1)}{2} f(i) \right) \]

By defining the second moment of the connection time distribution in the normal way, i.e., \( \overline{X^2} = \sum_{i=1}^{S} i^2 f(i) \), the above equation can be written
\[ R = \left( 1 - \frac{N - 1}{M} B \right) \left( \bar{X} + (1/r - 1) P_{\text{min}} + \frac{(N - 1) P_{\text{min}} R}{M} \frac{(\bar{X} - \bar{X}^2)}{2} \right) \quad (5) \]

The equations for \( P_{\text{min}} \) [(1)], for \( B \) [(4)], and the above equation form the MC model. They can be solved by iteration. In the experiments reported in the next section a fixed-point iteration on the value of \( R \) was used. Solution typically required 4 iterations (with a maximum of 8) when an initial value of \( R = r \) was used. Higher order iterations schemes could be used but were found unnecessary within the scope of our work. The value for memory bandwidth \( BW \), the probability of acceptance \( P_a \), and processor utilization \( U_p \), can be calculated from the following:
\[ BW = N(P_1 + B); \quad P_a = (1 - B') P_{\text{min}}, \quad U_p = 1 - \sum_{j=1}^{s} P_j \]

These equations follow from the definitions of Section II. It can be seen from (5) that \( R \) depends on, among other things, the inverse of \( \overline{X^2} \). Therefore, it follows from the above equations, that both \( BW \) and \( U_p \) depend on the inverse of \( \overline{X^2} \). Furthermore, it can also be seen from (5) that \( R \) is independent of \( S \). Thus, the underlying Markov chain of the MC model need not be finite. Finally, it is easy to show that the MC model properly subsumes the rate-adjusted model of Hoogendoorn (see [5], [10]), which corresponds to the case where \( X \) is a deterministic random variable of value 1, i.e., \( \bar{X} = 1 \) and \( \bar{X}^2 = 1 \).

IV. SIMULATION RESULTS

A simulation, written in SIMSCRIPT II.5, of multiprocessor systems operating according to the assumptions given in Section II was run for different probability mass functions \( f(i) \), and different values of \( r \). The results were compared to results calculated from the ER model and the MC model. The comparisons were made for different sized systems. Fig. 2 shows the results obtained from a 32 × 32 system (see [19] for additional results). The graphs compare \( BW \). The “error” shown in the figure was defined as
\[ \text{error} = \frac{\text{Model BW} - \text{Simulated BW}}{\text{Simulated BW}} \times 100. \]

Six different distributions for the connection time were used. All the distributions had the same expected value \( \bar{X} = 4.0 \) but their coefficient of variation \( C_v \) ranged from 0.0 (i.e., fixed connection time) to 2.0. As can be seen from the figure, both models gave results within 4 percent of the simulation for small \( C_v \). The MC model remained within this error bound, but the ER model showed a monotonic increase in error with increase in \( C_v \). In the case of \( C_v = 2.0 \) the error was as large as 50 percent. Similar results are obtained if \( P_a \) or \( U_p \) are compared.

The poor performance of the ER model, which continues to worsen as \( C_v \) increases beyond \( C_v = 2.0 \), confirms the importance of using the second moment of the connection time distribution in calculating \( BW \) (see [5]). The error in the MC model is due to assumption 3a) being used in place of assumption 3. As noted, the simulation works according to the assumptions of Section II, in particular assumption 3. The key difference between assumptions 3a) and 3) is that in 3a) blocked requests for memory need not be resubmitted to the memory from which they were previously blocked. This is clearly unrealistic, but our experimental evidence indicates that the effect on the quantities \( BW \), \( P_a \), and \( U_p \) is quite small — less than 4 percent error in all cases. Furthermore, as mentioned earlier, the relaxation of assumption 3) to that of 3a) makes possible a manageable model. Finally, the error is comparable with the empirical evidence reported in earlier work [3]–[5] that led to the assumptions of Section II as a phenomenological basis for the behavior of a large class of multiprocessors of the type discussed here.

Fig. 3 shows explicitly how \( BW \) varies with \( C_v \) while \( \overline{X} \) is held constant at 4.0. As can be seen, by just varying \( C_v \) from 0 to 2.0 the \( BW \) can drop by as much as 40 percent for high request rates (\( r = 1 \)). In fact, in the case where \( C_v = 2.0 \) the \( BW \) drops to the point where only 13 of the 32 memory modules are busy even where \( r = 1 \). This agrees with the interpretation, based on (12), that was made at the end of the last section where it was concluded that \( BW \) would decrease if \( \overline{X^2} \) (or \( C_v \)) increased. The most obvious consequence of \( BW \) depending on \( C_v \) in this way is that transfers between processors and memories should be restricted to fixed blocks, or nearly fixed blocks in which variations in size are rare, if maximum \( BW \) is to be achieved.

V. CONCLUSION

This correspondence has developed two discrete time models of the memory interference that occurs during memory access in a multiprocessor system when that access can have a variable duration. The first of these models, the ER model, is the simpler model and, according to comparisons to simulations, provides accurate...
Fig. 2. Comparisons to simulation results for a 32 × 32 system.

Fig. 3. The results of varying $C_r$ on $BW$. 
estimates of the values for $P_a$, $BW$, and $U_p$, if $C_a$ is small. The second of these, the MC model, is the more complex model but, according to comparisons to simulations, provides accurate estimates of the values for $P_a$, $BW$, and $U_p$, for a wide range of $C_a$. The ER model requires the values of $M$, $N$, $r$, and $X$ as inputs. The MC model requires, in addition, the value of $X^2$. The explicit dependence of the MC model on $X^2$ (and hence $C_a$) can be observed in (12). This was confirmed empirically; specifically, it was shown that $BW$ decreases with increase in $C_a$. The fact that the second moment is an important feature of memory interference should not be completely unexpected as the behavior of similar systems, e.g., networks of queues, also depend on the variance of underlying stochastic processes (in the case of queues, it is the variances of the interarrival time and service time).

ACKNOWLEDGMENT

The authors gratefully acknowledge comments and suggestions made by B. A. Makrucki.

REFERENCES