A MODEL FOR MEMORY INTERFERENCE IN MULTIPROCESSORS

Stanley Rabinowitz
Alliant Computer Systems Corporation
Littleton, MA 01460

Abstract. A model is described and analyzed for a multiprocessor shared memory system in which each memory bank can service a fixed number of access requests per cpu cycle.

If n processors simultaneously request data from a common shared memory, it is usually not possible for all the requests to be satisfied at the same time. This is because the memory system usually places a limit on the number of requests that it can service at any given time. A typical method for allowing multiple requests to be satisfied is to divide the memory into m banks each capable of satisfying requests independently. The memory banks are usually interleaved, so that requests to successive memory locations will be serviced by successive memory banks.

Even if $m = n$, full memory bus bandwidth cannot usually be achieved. This is because of the fact that if n processors each make a memory request at random, it is unlikely that the n requests will all be to different memory banks. In fact, the probability that the n requests go to all n banks is just the number of permutations of Z_n divided by the number of mappings of Z_n into Z_n, where Z_n represents the set of integers from 1 to n. This

probability is

$$\frac{n!}{n^n}$$

which, as n gets large, approaches $\sqrt{2\pi n}/e^n$ by Stirling's Formula.

One solution is to make $m > n$. Since the cost increases with the number of memory banks, it is important to know how the average memory bandwidth will be affected by the values of m and n. Several mathematical models have appeared in the literature to predict the behavior of such systems [1–5,8,11–13]. The model that is closest to the behavior of the machine we are interested in (the Alliant FX/8 multiprocessor) is the model that Hoogendoorn [8] calls the Uniform Static-Access Matrix. In this model, each processor accesses each memory bank with equal probability $(1/m)$. The memory requests are considered to be random and independent. If a memory bank gets more than one request, it services precisely one of the requests. The other requests are rejected and are not queued up. There is no assumption that the requesting processor will make the same request on the next cpu cycle. (This is clearly a deficiency in this model.)

Ravi [13] showed that the expected number of requests that are serviced is

$$E(m,n) = \sum_{k=1}^{\min(m,n)} \frac{k \cdot k! S(n,k) \binom{m}{k}}{m^n} \tag{1}$$

where $S(n,k)$ denotes the Stirling numbers of the second kind (see [6, p. 121] or [10, p.

65]), defined by the formula

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^k\binom{k}{i}(k-i)^n \ .$$

It was later pointed out by several authors [5, 8, 14], that formula (1) can be simplified to

$$E(m,n) = m\left[1-\left(1-\frac{1}{m}\right)^n\right] \ . \tag{2}$$

It was pointed out by Chang, Kuck, and Lawrie [5] that if the ratio $s = n/m$ is fixed, then this function is asymptotically linear, because

$$\lim_{n\to\infty}(1-\frac{1}{m})^s = \frac{1}{e^s}$$

so that

$$\lim_{n\to\infty} E(m,n) = n\left(1-\frac{1}{e^s}\right)/s \ .$$

Another deficiency of the model is that it assumes that each memory bank can service only one request per cpu cycle. On the Alliant FX/8, the memory bus is twice as fast as the cpu, so that each memory bank can service up to two memory requests per cpu cycle.

We thus wish to extend the model to assume that each memory bank can service up to r requests per cpu cycle. The result of this enhanced model can be summarized by the following theorem.

Theorem. At a given time, if n processors make independent and random memory requests to m memory banks (with each memory bank being equally likely) and if each bank can satisfy at most r requests at a time, then the expected number of requests that are satisfied is

$$E(m,n,r) = mr - m\sum_{k=0}^{r-1}(r-k)\binom{n}{k}\left(\frac{1}{m}\right)^k\left(1-\frac{1}{m}\right)^{n-k} . \tag{3}$$

Proof. Look at the situation from the viewpoint of one of the memory banks (say bank j). Let $\Pr(k)$ denote the probability that this bank gets k memory requests. Since the probability that any given request goes to a particular bank is $1/m$, the probability that a given request does not go to a particular bank is $1-1/m$. We want k requests to go to bank j and all the other $n-k$ requests to go to other banks. There are $\binom{n}{k}$ ways to pick k lucky processors to be privileged to access bank j. Their requests indeed go to this bank with probability $(1/m)^k$. The probability that the requests from the $n-k$ other processors do not go to this bank is $(1-1/m)^{n-k}$. Thus

$$\Pr(k) = \binom{n}{k}\left(\frac{1}{m}\right)^k\left(1-\frac{1}{m}\right)^{n-k} .$$

The expected number of requests that will be serviced by bank j is

$$E_j(m,n,r) = \sum_{k=0}^{r-1} k\Pr(k) + \sum_{k=r}^{n} r\Pr(k)$$

since when r or more requests come in, only r are serviced. Since

$$\sum_{k=r}^{n}\Pr(k) = 1 - \sum_{k=0}^{r-1}\Pr(k) ,$$

we can rewrite this as

$$E_j(m,n,r) = \sum_{k=0}^{r-1} k \Pr(k) + r\big[1 - \sum_{k=0}^{r-1} \Pr(k)\big] = r + \sum_{k=0}^{r-1} (k-r) \Pr(k) \ .$$

Substituting in the value of $\Pr(k)$ gives

$$E_j(m,n,r) = r + \sum_{k=0}^{r-1} (k-r) \binom{n}{k} \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{n-k} \ .$$

Since the expected number of services performed by each memory bank is the same, $E(m,n,r)$ is just m times the above expression. ∎

It is interesting to note that formula (3) can be expressed in the form

$$E(m,n,r) = \frac{1}{(r-1)!} D_m^{r-1} m^r \left[1 - \left(1 - \frac{1}{m}\right)^n\right] \tag{4}$$

where $D_x^k f(x) \equiv f^{(k)}(x)$ denotes the kth derivative of $f(x)$. This follows from the generalization of Halphen's formula [7, p. 161], which states that

$$\left[F\left(\frac{1}{x}\right) G(x)\right]^{(N)} = \sum_{k=0}^{N} (-1)^k \left(\binom{N}{k}\right) \left(\frac{1}{x}\right)^k F^{(k)}\left(\frac{1}{x}\right) \cdot \left[\frac{G(x)}{x^k}\right]^{(N-k)} \ .$$

Letting $F(x) = 1 - (1-x)^n$, $G(x) = x^r$, and $N = r - 1$ shows the equivalence of (3) and (4).

Although formula (4) is more compact, it comes from a formal identity and there does not appear to be any physical reason why derivatives occur in the solution to our problem.

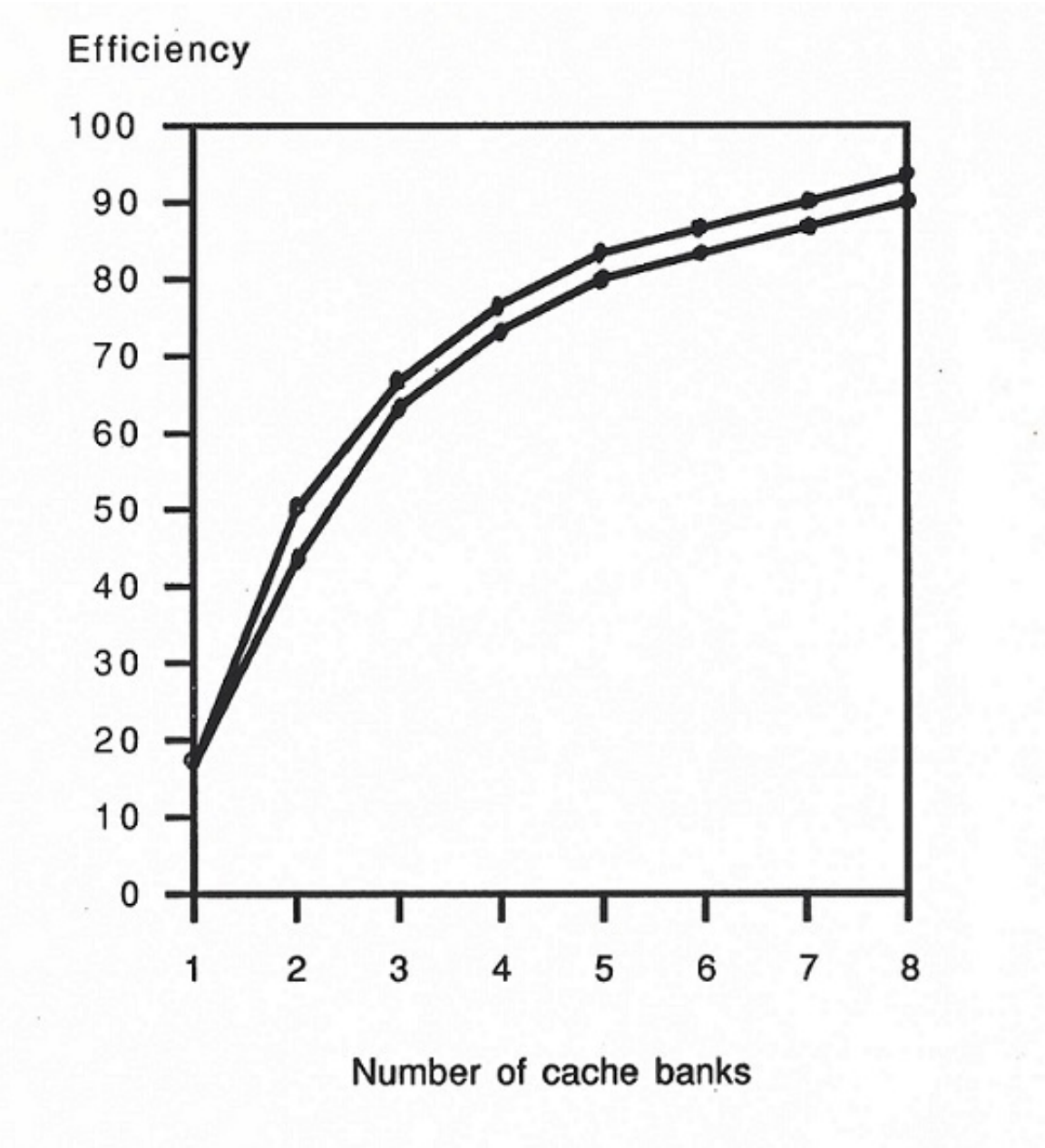
The model used by formula (3) does not precisely predict the performance of the Alliant FX/8. This is because it fails to take into consideration effects caused by memory requests

that get queued up when a processor stalls. The model assumes that each memory request is independent of previous requests. This is not true. If a memory request does not succeed on a given cpu cycle (because more than r requests were made to the target memory bank), then the stalled processor will repeat that request again on the next cpu cycle.

A detailed simulation of the FX/8 memory architecture was implemented by Jacobs [9]. This simulation allows varying the parameters m, n, and r. Both the model and the simulation confirm that in a choice between doubling m, the number of memory banks, and doubling r, the number of requests a given bank can handle, the better bandwidth is achieved by doubling r.

Figure 1 shows the efficiency predicted by both the model and the simulation when $r = 2$ and $n = 8$ as m varies from 1 to 8.

The efficiency is defined as the ratio of the average number of memory requests serviced per cpu cycle divided by the theoretical maximum achievable (n). The ratio is multiplied by 100 to express the result as a percentage. This chart shows that the mathematical model overestimates the memory performance by as much as 5%.


Efficiency
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
Number of cache banks

References

1. D. H. Bailey, "Vector Computer Memory Bank Contention," *IEEE Transactions on Computers*, C-36 (1987), 293–298.

2. F. Baskett and A. J. Smith, "Interference in Multiprocessor Computer Systems with Interleaved Memory," *Communications of the ACM*, 19 (1976), 327–334.

3. F. A. Briggs and E. S. Davidson, "Organization of Semiconductor Memories for Parallel-Pipelined Processors," *IEEE Transactions on Computers*, C-26 (1977), 162–169.

4. D. P. Bhandarkar, "Analysis of Memory Interference in Multiprocessors," *IEEE Transactions on Computers*, C-24 (1975) 897–908.

5. D. Y. Chang, D. J. Kuck, and D. H. Lawrie, "On the Effective Bandwidth of Parallel Memories," *IEEE Transactions on Computers*, C-26 (1977) 480–490.

6. D. I. A. Cohen, *Basic Techniques of Combinatorial Theory*, John Wiley & Sons, New York, 1978.

7. L. Comtet, *Advanced Combinatorics*, D. Reidel Publishing Company, Boston, 1974.

8. C. H. Hoogendoorn, "A General Model for Memory Interference in Multiprocessors," *IEEE Transactions on Computers*, C-26 (1977) 998–1005.

9. H. R. Jacobs, Alliant Computer Systems Corporation, personal communication.

10. D. E. Knuth, *The Art of Computer Programming, Volume 1, Fundamental Algorithms*, Addison-Wesley, Reading, MA, 1969.

11. D. E. Knuth and G. S. Rao, "Activity in an Interleaved Memory," *IEEE Transactions on Computers*, C-24 (1975) 943–944.

12. J. H. Patel, "Analysis of Multiprocessor with Private Cache Memories," *IEEE Transactions on Computers*, C-31 (1982) 296–304.

13. C. V. Ravi, "On the Bandwidth and Interference in Interleaved Memory System," *IEEE Transactions on Computer*, C-21 (1972) 899–901.

14. W. D. Strecker, *An Analysis of the Instruction Execution Rate in Certain Computer Structures*, Ph.D. dissertation, Carnegie-Mellon University, 1970.