U.S. DEPARTMENT OF COMMERCE • National Institute of Standards and Technology

National Computer and Telecommunications Laboratory

NISTIR 88-3857

On the Performance Evaluation and Analytic Modeling of Shared-Memory Computers

James Nechvatal

U. S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
National Computer and Telecommunications Laboratory
Advanced Systems Division
Gaithersburg, MD 20899

September 1988

CMRF

COMPUTER MEASUREMENT
RESEARCH FACILITY
FOR HIGH PERFORMANCE
PARALLEL COMPUTATION

Partially sponsored by the
Computer Systems Technology Center,
Office of Export Administration,
U.S. Department of Commerce.

ON THE PERFORMANCE EVALUATION AND ANALYTIC MODELING OF SHARED-MEMORY COMPUTERS

James Nechvatal

Advanced Systems Division
National Computer and Telecommunications Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

Preparation of this report was sponsored in part by the
Computer Systems Technology Center
Office of Export Administration
Department of Commerce
Herbert C. Hoover Building
14th and Constitution Avenue, N.W.
Washington, DC 20230

The work reported here was performed at the National Institute of Standards and Technology (NIST), an agency of the U.S. Government, and is not subject to U.S. copyright. The identification of commercial products in this paper is for clarification of specific concepts. In no case does such identification imply recommendation or endorsement by NIST, nor does it imply that the product is necessarily the best suited for the purpose.

U.S. Department of Commerce, C. William Verity, Jr., Secretary

National Institute of Standards and Technology, Ernest Ambler, Director

September 1988

ABSTRACT

We survey and discuss analytic performance models for shared-memory computers. Our purpose is to assess these models to determine if they are relevant to the task of assisting the Office of Export Administration in establishing guidelines for exports of computers. The focus of this study is on the assumptions made by modelers regarding interconnection networks and memory access patterns exhibited by applications. The great majority of models concentrate on analyzing a small collection of networks. Furthermore, most modelers make strong assumptions regarding the independence and uniform distribution of memory accesses. The consequences of these assumptions in the resulting models are noted.

KEY WORDS

computers; export; memory; models; networks; performance

TABLE OF CONTENTS

1. Introduction.

The Office of Export Administration of the Department of Commerce has requested the assistance of the National Computer and Telecommunications Laboratory, NIST, in establishing guidelines for export of computers. One possible avenue is the use of analytic models to predict performance of computers (alternatives such as simulation and benchmarking are not treated here). We examine a number of models which have been proposed for the performance evaluation of parallel computers in which processors share common memory. We are not concerned here with the details of models; rather, we are interested in the question of whether such modeling efforts are primarily of interest to theoreticians and machine designers, or whether they can serve as accurate and reliable predictors of performance of real machines on real applications.

We assume that the memory of a shared-memory computer is partitioned into modules; processors access the modules via some form of interconnection network. Alternatively, memory may be shared but distributed, with processor/memory pairs connected by an interconnection network. In either case the main factor in performance is the efficiency of the network in permitting processors to access memory. In Sections 2 and 3 we review some classes of interconnection networks. In Sections 4-8 we survey various performance models, and in Section 9 we return to the question of the value of these models in the present context.

In passing we remark that hierarchical systems, in which processors are grouped into clusters sharing cache or common memory, are not covered here.

2. Interconnection networks.

By interconnection network we refer to any structure connecting a set of sources to a set of sinks. Typically the sources are processors and the sinks are memory modules; this is the case we will refer to in the following. Alternatively, however, both sources and sinks could be processor/memory pairs. Four major categories of networks are:

A. crossbar

B. multistage network

C. multiple bus

D. single bus

Crossbars provide complete interconnection; i.e. it is possible to concurrently connect any n sources to any n sinks. Their expense, however, is prohibitive for large n. Single buses are inexpensive, and have been heavily used to date; however, they permit connection of only one (source,sink) pair at one time, and hence are suitable only for small numbers of sources. The other two categories are intermediate with respect to both expense and connection capability. Multiple bus systems consist of several buses with each bus connected to all sinks. They may be complete (each bus connected to all sources) or partial (each source connected to a subset of buses). Multistage networks have been intensively studied, and many references can be found in the bibliography (e.g. [39],[51],[97]). It seems likely that multistage networks will grow increasingly important as systems incorporate larger numbers of processors. Thus we review them in detail in the next section.

3. Taxonomy of multistage networks.

Multistage networks consist of switches which route communications between sources and sinks. The switches are arranged in several stages, with the sources connected to the inputs of the first stage, and the sinks connected to the outputs of the last stage. In intermediate stages, the outputs of a stage are connected to the inputs of the next stage. Communication consists of a request for access by a source to a sink. In practice a successful request is followed by transmission of data from sink to source, but for simplicity we refer only to requests for access by sources. Multistage networks can be classified according to features such as the following (not all terminology is universal):

I. transmission mode and capability

  A. packet-switching

    i. blocking

      a. blocked request queued

      b. blocked request discarded

    ii. nonblocking

  B. circuit-switching

    i. blocking

      a. queued

      b. rejected

    ii. non-blocking

      a. strict

      b. wide-sense

    iii. rearrangeable

II. Paths from a source to a sink

  A. unique path

  B. multipath

III. Switch control

  A. central

  B. distributed

C. stage

IV. Synchronicity

A. synchronous

B. asynchronous

V. Permutation realization capability

VI. Switch boxes

A. fan-in / fan-out

i. rectangular (k x n)

ii. square (n x n)

B. functions

i. interchange

ii. broadcast

C. queueing of requests

i. buffered

ii. unbuffered

D. arbitration policy

i. fair

ii. biased

VII. Topology

A. omega

B. indirect binary n-cube

C. generalized cube

D. banyan

E. delta

F. baseline

G. data manipulator

H. flip

I. other

Packet-switching networks ([23],[24],[33]) route packets individually from source to sink without establishing a fixed connection; i.e. links are used and then relinquished. They may be non-blocking; i.e. a packet can always be routed without conflict. More commonly they are blocking; i.e. two or more packets may arrive on the input wires of a switch requesting the same output wire. If the network is buffered, packets may be queued; otherwise one packet will be selected and the rest lost, presumably requiring resubmission.

Circuit-switching networks ([17],[103],[107]) set up a fixed connection between source and sink for the duration of a message; i.e. links are held rather than relinquished. Again the network may be blocking; but if a conflict occurs at a box, a partially established circuit may need to be torn down and a retry made by the issuing source. In [17] the non-blocking case is split: strictly non-blocking means circuits can be constructed at will without fear of blocking; wide-sense means a particular algorithm must be used to avoid conflicts. Rearrangeable means that existing circuits can be restructured to accomodate a new circuit. A comparison of circuit and packet switching is given in [98].

A network generally has at least one path from each source to each sink. Omega networks [64] and topologically equivalent networks (equivalent under relabeling of sources and sinks) have unique paths. Adding extra stages to such networks yields multiple paths from source to sink ([23],[24],[44],[60],[65],[98],[100]). Alternatively, extra paths can be produced by changing from 2 x 2 to k x k switch boxes [84].

The most common control mechanism for switches is distributed; i.e. each box is self-controlled. This is connected with data-routing [46]. Often self-routing is employed, via destination tags ([23],[24],[39],[64],[112]). By inspecting these tags, forwarded with packets, boxes are able to route packets to their destination. Central control is also a possibility, as is stage control [5].

A network may be synchronous, i.e. it may accept packets only at the beginning of memory cycles. A synchronous packet-switched network can be pipelined: the stages of the network form a pipeline, so that sets of packets can be present concurrently at all stages. Packets are admitted to the first stage at the start of each cycle. Pipelining cannot be used with circuit-switching; a circuit occupies all stages simultaneously.

Permutation capabilities refer to the case where the sources all submit requests to different sinks. If an equal number of sources and sinks are present this forms a permutation. The latter is realizable if the requests can be routed without conflict. Realization capabilities have been studied for the omega network ([27],[46],[83]) and its equivalents, as well as its single-stage parent, the shuffle-exchange network ([61],[62],[114]). Capabilities of other networks have been studied ([21],[103],[106]) as well, but the omega-like case is most interesting.

Determinations of capabilities for extra-stage networks are challenging ([1],[44],[65]).

Switch boxes are usually crossbars, connecting each input to each output. In the 2 x 2 case, broadcast capability is sometimes assumed: an input can be sent to both outputs. Two inputs are never sent to one output. Also in the 2 x 2 case, there are 4 possible settings: straight (1-1,2-2), interchange (1-2,2-1) or broadcast ((1-1,1-2) or (2-1,2-2)). This is called four-function capability. If broadcast is excluded it is called a two-function box.

A conflict occurs at a switch when two or more inputs must be routed to the same output. In a buffered network a queue provides temporary storage for blocked requests [58]; otherwise one request is selected for forwarding and other requests must be discarded or resubmitted by the sender. Conflict resolution schemes are normally fair (random selection) but other arbitration policies are occasionally used ([10],[20],[101]).

The most well-known networks are based on the shuffle ([19],[61]), a permutation which sends an integer whose binary representation is (i_{n-1} , ... ,i_1 ,i_0) to the integer with representation (i_{n-2} , ... ,i_0 ,i_{n-1}). Individual switches perform, in interchange mode, exchanges, sending (i_{n-1} , ... ,i_1 ,i_0) to (i_{n-1} , ... ,i_1 ,$1-i_0$). The omega network [64] is a multistage shuffle/exchange network connecting N inputs to N outputs. It is composed of $\log_2$ N stages of N/2 switches, each 2 x 2. Between each pair of stages, routing outputs of a stage to inputs of the next stage is done by a shuffle. Networks equivalent to the omega include the flip [5], indirect binary n-cube [89], modified data manipulator and baseline [113], generalized cube [99] etc. All of these connect N sources to N sinks with a unique path from source to sink; they all have $\log_2$ N stages with each stage consisting of N/2 switches, each a 2 x 2 box. The differences between them are in the control structures and the precise class of permutations which they can realize. See ([100],[112],[113]) for more details on these.

The omega-like networks have been generalized in various ways. Deltas [85] use a x b switch boxes to connect a^n sources to b^n sinks; the 2 x 2 case yields the omega. Banyans [45] are characterized by the property of a unique path between each (source,sink) pair; they include deltas and in particular omegas (square SW-banyans). On the other hand, multipath networks, as noted earlier, have been obtained by adding stages while keeping the switch box or expanding switch boxes while retaining the number of stages.

4. Performance models.

As noted in the Introduction, the primary determinant of the performance of shared-memory computers is their capability for concurrently connecting sources and sinks. A common measure of this capability is the bandwidth of the interconnection network. Typically bandwidth refers to the number of memory requests accepted in a memory cycle ([11],[12],[13],[18],[49],[86]) or more generally the number of memory modules accessed concurrently. In multiple bus systems bandwidth may also be characterized in terms of the number of active buses in a bus cycle ([77],[78]); typically these characterizations do not conflict since it is assumed that a transaction between processor and memory can always be completed in a bus cycle. Similarly, if networks have cycles it is usually assumed that their operation is synchronous with modules. In asynchronous single bus systems, bus utilization is a measure of performance [3].

Performance models typically ignore the management of mutilevel memory. Accesses to private caches by a processor are generally considered part of computation rather than communication; systems are modeled as a collection of sources (processors or caches) sending requests (for access) to a collection of sinks (memory modules). Requests are serviced first by the interconnection network and then, if the request is granted, by the module. Two forms of contention arise: over the use of the network, and for concurrent access to a module. We concentrate here on studying the assumptions made by various modelers concerning the issuing of requests by processors and the disposition of those requests in passage through the network. In particular it is of interest to note what occurs when two requests conflict.

5. General assumptions.

There are few general assumptions upon which authors concur on this subject. An exception is that virtually all models assume that processors and modules are homogeneous. Furthermore, a common convention is that a processor is always in one of three states ([49],[52],[71],[72],[73],[75],[76]):

1. running, i.e. working in private memory.

2. waiting for access to a module.

3. accessing a module.

However, many authors assume a processor is never waiting; this is because it is assumed that when a processor's request for memory access is blocked, it simply issues another request. We return to this later. Also, a few authors assume that a processor is never running (i.e. it spends all of its time issuing requests).

Some authors define a measure supplementary to memory bandwidth, namely processor utilization ([49],[52],[73],[75],[87]), i.e. the number of processors in a running state during a memory cycle. Of course these measures are interconnected.

Many authors assume the network and memory modules are synchronous: processors issue requests only at the beginning of memory cycles ([4],[6],[7],[8],[9],[11],[12],[13],[18],[22],[29],[30],[32],[34],[56],[59],[63],[74],[76],[79],[80],[82],[85],[86],[90],[92],[93],[96],[104],[105],[109],[110],[118],[119]) or occasionally bus cycles ([75],[77],[78]). In the case of multistage interconnection networks, in the synchronous, packet-switched case the network is typically pipelined. Less frequently the system is allowed to be asynchronous ([3],[26],[42],[47],[48],[52],[53],[55],[69],[70],[72],[73],[108],[111],[115]). The asynchronous case arises in particular in analyses of real-time systems ([3],[47],[48],[69]).

Multistage networks are usually blocking. Packet-switching is often assumed ([20],[22],[23],[24],[32],[57],[58],[59],[81],[119]) but circuit switching is also common ([26],30],[40],[66],[82],[87],[88],[104],[105],[110]); some models treat both ([31],[56]). In packet-switching, the default is messages consisting of single packets; but occasionally the multipacket case is considered [31].

Blocking networks may be buffered ([20],[24],[57],[81]) or unbuffered ([22],[59]); sometimes both are treated ([31],[32],[56],[58],[119]). Buffer size may be a parameter [32]. In buffered systems requests are normally queued in the event of conflict; but then full buffers are another form of blocking, unless queues are infinite

[81].

Many analyses assume that processors have private memories; in some cases the role of private cache is emphasized ([15],[16],[35],[36],[53],[54],[67],[79],[81],[87],[88],[115]). Caches are occasionally shared ([37],[116], [117]).

Other issues such as switch control [32] and data routing ([12],[20],[30],[82]) are occasionally addressed. Also, in the event of conflict at a switch or over bus usage, normally one processor is selected randomly; occasionally other arbitration policies are used ([10],[20],[68],[101]). Multistage networks can also be differentiated on the basis of topology or shape (square, rectangular). However, the most important specifications (and the ones most frequently made clear) are discussed in the next section.

6. Distribution and treatment of processor requests.

Distribution of requests by processors is a specification which heavily influences bandwidth. In the event of conflict between requests (e.g. at a switch or over bus usage) the disposition is also a major factor. A partial taxonomy of assumptions made by modelers is as follows:

I. Let p_{ij} be the probability that processor i issues a request to module j, with m modules present. Then possibilities include:

A. p_{ij} = 1/m for all i and j; this is the uniform case in which requests to modules are equiprobable ([4],[6],[8],[12],[14],[18],[20],[22],[26],[30],[31],[32],[38],[42],[53],[55],[56],[59],[63],[70]-[79],[81],[82],[85]-[88],[92],[93],[104],[105],[109],[110],[115],[118],[119]).

B. for some a: for each i, for some j_i we have p_{ij} = a if j = j_i and (1-a)/(m-1) otherwise; i.e. each processor has a favorite memory ([9],[10],[11],[13],[29],[50],[80]).

C. for some a and k: for all i, p_{ij} = a if j = k and (1-a)/(m-1) otherwise; i.e. all processors have the same favorite (hot) memory ([50],[90]).

D. for some a: if processor i has just referenced module k, then in the next cycle p_{ij} = a if j = k and (1-a)/(m-1) otherwise (local reference model) ([96],[101]).

E. the {p_{ij} } are independent random variables ([41],[43],[102]).

F. for each j: for some p_j , p_{ij} = p_j for all i ([34],[108]).

II. If a processor's request is blocked:

A. the request is discarded ([7],[9],[12],[13],[14],[22],[26],[29],[56],[59],[77],[78],[80],[85],[86],[90],[104],[109],[110]).

B. the request is resubmitted ([4],[8],[18],[34],[38],[74],[76],[79],[82],[87],[88],[105],[118]).

C. A or B ([30],[66]).

III. In a synchronous system, in each memory cycle, a processor:

A. issues a new request with probability p, p fixed, $0 < p < 1$ ([4],[7],[9],[10],[12],[13],[22],[29],[30],[34],[38],[41],[56],[59],[63],[68],[74]-[78],[82],[85],[86],[104],[109],[110],[118],[119]).

B. issues a new request with probability one ([8],[18],[32],[96],[101]).

IV. All requests by processors are independent of both previous requests and also requests issued concurrently by other processors ([7],[8],[9],[11]-[14],[20],[22],[29],[31],[32],[38],[50],[59],[63],[77]-[80],[82],[86],[87],[88],[90],[93],[104],[109],[110],[115],[119]).

V. In an asynchronous system, time between requests and time for processors to access modules may vary, but both have the same mean for all processors ([26],[42],[52],[53],[55],[70],[71],[72],[73],[75],[108],[115]).

It is important to observe, as noted in ([8],[38],[79],[82]), that (II-B) and (IV) are contradictory; this simplifies subsequent analyses. The same contradiction occurs in ([87],[88]) but is not explicitly noted. Implicitly this issue is raised whenever (II-A) is invoked as well: it is unrealistic to assume that a processor whose request is discarded will forgive and forget and issue a new independent request, as noted in ([26],[29],[77],[78],[80],[85],[86]). In other words, (IV) is king; rarely do authors try to defy it, since the resulting analysis is complicated if they do. For example, (IV) is violated by (I-D). The latter is reasonable given the well-known spatial and temporal locality of reference exhibited by uniprocessors; nonetheless authors avoid it scrupulously. In ([49],[81],[82],[91],[96],[101],[110]) it is noted that one or more of (I-A), (III-B) and (IV) are also typical of assumptions which are unrealistic in practice but which simplify analysis.

It should be noted that these simplifications are akin to ignoring issues such as data dependency in parallelizing loops, or branching in instruction lookahead.

7. Further notes.

As might be expected, strong assumptions and simple architectures combine to produce the most tractable frameworks, and also the easiest to draw conclusions about. For example, authors love crossbars ([9]-[13], [16],[29],[34],[38],[40],[41],[53],[63],[74],[76],[80],[86],[87],[88],[90],[92],[93],[94],[96],[101],[102],[104],[118]). However, barring new technological developments (e.g. [95]), large crossbars are unlikely ever to be implemented because of their prohibitive cost. The crossbar has the unique advantage to both processors and modelers of eliminating one of the two sources of contention encountered by requests, namely over use of the network. The implication for modelers is that if this architecture is combined with an appropriate set of assumptions from the preceding section, the task of modeling becomes trivial. For example, if p processors make independent, identically distributed requests to m modules with no blocking possible, the expected bandwidth is simply $m(1-(1-1/m)^p)$. This is an extreme example of how strong assumptions can produce simple results; but the value of the latter is highly questionable.

Multiple buses are, in general, not as easy to analyze as crossbars, but are still quite tractable ([8],[10],[11],[25],[28],[29],[36],[41],[42],[49],[52],[55],[63],[68],[70],[73],[75],[77],[78],[79],[108],[109],[115]). However, multiple bus systems are as rare as large crossbars. Single buses ([47],[48],[69],[71],[72],[111]) occur much more frequently and are relatively easy to analyze; however it is well-known that single buses support only small numbers of processors.

Multistage interconnection networks are difficult to analyze because events occur at each stage. They are nonetheless treated frequently ([9]-[13],[22],[26],[30],[31],[32],[40],[43],[56],[57],[59],[60],[66],[67],[81], [82],[85]-[88],[104],[105],[110],[119]). The role of independence in simplifying analyses should be noted in this regard, as we remarked in the last section. Many analyses also exploit uniqueness of paths from source to sink; a few authors permit multipaths ([23],[24],[82],[110]). In [20] single-stage networks are analyzed; in [58] both single-stage and multistage networks are considered.

There are few simple conclusions which are drawn from most studies. A random selection follows. In [13] it is noted that if processors have favorite modules that are accessed frequently, multistage networks have about the same (high) bandwidth as crossbars. Adding extra stages produces similar effects: wait delays in queues are reduced [24]. In [32] buffers of different sizes are considered, including zero; a conclusion is that diminishing gains are obtained from large buffers. An implication is that an assumption of infinite buffers is not inaccurate, and simplifies modeling. In circuit switched networks, tearing down partial circuits in event of a block may be preferable to holding them [66]; however, the opposite conclusion is reached in [30]. In [31] it is shown that packet-switching may actually out-perform circuit-switching for long messages. In a multiple bus system, one bus for two processors gives nearly the same bandwidth as a crossbar (one bus per processor) ([63],[77],[78]).

In many cases the significance of results is not clear. In particular, few authors show that their analyses are applicable even to a single real application/architecture pair.

8. A case study.

A study in [56] illustrates some strengths and weaknesses of the modeling efforts catalogued in this report. This study has been cited frequently in the literature, and it is one of the few papers on multistage networks which contain simple conclusions. In one section the authors assume:

A. a banyan network (multistage with unique path from each source to each sink).

B. k x k switches.

C. unbuffered.

D. k^n inputs, k^n outputs and n stages (square banyan).

E. synchronous network.

F. processors act as independent, identically distributed random processes in issuing requests (more briefly, requests are independent).

G. a processor generates, with probability p, a packet in each cycle.

H. a processor's packets are directed to all modules with equal probability.

I. when several packets conflict at a switch, one is fowarded and the rest discarded.

As may be seen from Section 6, many of these assumptions are typical, but this particular set is stronger than most as a group. To explore the consequences, let p_j be the probability that an input line to a switch at stage j has a packet. Then it follows easily that

$$p_{j+1} = 1 - (1 - \frac{p_j}{k})^k \qquad , \; j \geq 0$$

$$p_0 = p$$

It follows that the probability that a packet is not blocked is p_n ; if the packet is replaced by a request to set up a circuit, p_n is the probability that a circuit is established. The simplicity of this result, and its simultaneous applicability to both circuit and packet switching, raise the question as to whether such bandwidth calculations have real significance. The answer is yes, but only if an application has the courtesy to exhibit the required behavior. In particular, requests must be issued independently and be directed randomly to all modules, and when their requests are discarded, processors must not resubmit their requests and upset the independence assumption.

It should be noted that the above work in [56] is based on earlier work in [86]. In particular, this strong set of assumptions is essentially carried over from [86]. Furthermore, this work is continued in [22] and [59]; again these strong assumptions are made. This is an example of a series of increasingly precise conclusions being drawn about a situation whose existence is debatable.

9. Conclusions.

In inspecting the modeling efforts surveyed here, some interesting observations emerge. For example:

1. Architectures are selected for modeling on a somewhat random basis. There is roughly an equal split between crossbar, multiple bus and multistage, despite the paucity of the first two in present or planned machines.

2. Modelers lag behind machine designers. For example, a recent trend has been toward clustered systems; [2] is one of the few modeling efforts to address this approach to shared-memory architectures.

3. In many instances, and often by authors' admissions, assumptions about systems and requests issued by processors are made for the express purpose of tractability. Behavior of real applications on real machines is often an extraneous consideration.

4. In particular, assumptions made by modelers usually center around independence and random distribution of requests for memory access, despite the well-known theory of spatial and temporal locality for uniprocessors which, if extrapolated to parallel machines, would invalidate these assumptions.

5. Most analytical models of performance are statistical in nature, explaining the strong bias towards independence and uniformity. Models with this orientation implicitly assume a population of processes lacking a high degree of interaction which would produce dependence and nonuniformity in accessing patterns.

6. Effects of synchronization are rarely considered. It has been noted that synchronization mechanisms such as global locks and counters produce very nonuniform access patterns.

It is interesting to note that some of the older studies ([7],[14],[91]) placed considerable emphasis on connecting models and real application programs, e.g. in regard to storage patterns. For example, it was noted that low- and high-order interleaving have different effects on locality of reference. More recent studies have become increasingly abstract, with a consequent weakening of the correlation between models and the behavior of real programs. Some exceptions should be noted: for example, in [38] a model is developed for synchronous iterative computations arising in connection with such problems as solution of linear systems and partial differential equations. This framework is then interconnected with performance evaluation of various possible interconnection networks. In [27] a study is made of the connection between access capabilities of networks and applications such as Fast Fourier Transform and grid computations. A particularly interesting exposition is found in [81], where general models of both program structure and network performance are developed. The authors also note that many assumptions made by modelers are unrealistic. In particular, they note that many applications involve considerable synchronization, producing frequent access to a few memory modules. In Section 6 we noted that treatment of the cases of favorite and hot memories has been nearly nonexistent.

Unfortunately the three preceding papers are among the relatively few instances in which performance modeling of shared-memory systems and the behavior of actual applications are interconnected. This leaves an enormous gap to be filled. Much more work needs to be done in examining the profiles of parallel applications with regard to patterns of memory access. Furthermore, at this point a characterization of "typical" interconnection schemes does not exist, although as we have noted it appears likely that multistage networks will be prominent. Pending accumulation of this type of information on applications and architectures it is difficult to ascertain whether the models discussed here are accurate predictors of performance.

One major inhibitor to date in the accumulation of this needed information is the distribution of architectures. Most current nonhierarchical shared-memory machines are based on single buses, which do not support concurrent memory access. On such systems the issue of bandwidth, as defined in Section 4, does not even arise per se. Thus there is at present an almost total discrepancy between common architectures and modeled architectures. It appears that this situation will change in the near future, spurred by support from sources such as the Defense Advanced Research Projects Agency. When shared-memory architectures begin to proliferate, presumably the preceding information gap will begin to close, and it should become much clearer as to which of the preceding models, if any, have practical significance. Until then the lack of knowledge of profiles of typical applications and machines makes it difficult to assess these models with a high degree of confidence.

BIBLIOGRAPHY

1. G. B. Adams III and H. J. Siegel, "On the number of permutations performable by the augmented data manipulator network," IEEE Transactions on Computers, Vol. C-31, No. 4, April 1982, pp. 270-277.

2. D. P. Agrawal and I. E. O. Mahgoub, "Performance analysis of cluster-based supersystems," Proceedings of the First International Conference on Supercomputing Systems, St. Petersburg, FL, December 16-20, 1985, pp. 593-602.

3. W. L. Bain, Jr. and S. R. Ahuja, "Performance analysis of high-speed digital buses for multiprocessor systems," 8th Annual International Symposium on Computer Architecture: Conference Proceedings, 1981, pp. 107-133.

4. F. Baskett and A. J. Smith, "Interference in multiprocessor computer systems with interleaved memory," Communications of the ACM, Vol. 19, No. 6, June 1976, pp. 327-334.

5. K. E. Batcher, "The flip network in STARAN," Proceedings of the 1976 International Conference on Parallel Processing, pp. 65-71.

6. D. P. Bhandarkar, "Analysis of memory interference in multiprocessors," IEEE Transactions on Computers, Vol. C-24, No. 9, September 1975, pp. 897-908.

7. D. P. Bhandarkar, "Some performance issues in multiprocessor system design," IEEE Transactions on Computers, Vol. C-26, No. 5, May 1977, pp. 506-511.

8. L. N. Bhuyan, "A combinatorial analysis of multibus multiprocessors," Proceedings of the 1984 International Conference on Parallel Processing, pp. 225-227.

9. L. N. Bhuyan, "An analysis of processor-memory interconnection networks," IEEE Transactions on Computers, Vol. C-34, No. 3, March 1985, pp. 279-283.

10. L. N. Bhuyan, "Effect of arbitration policies on the performance of interconnection networks," Proceedings of the 1986 International Conference on Parallel Processing, St. Charles, IL, August 19-22, 1986, pp. 4-7.

11. L. N. Bhuyan, "Analysis of interconnection networks with different arbiter designs," Journal of Parallel and Distributed Computing, Vol. 4, No. 4, August 1987, pp. 384-403.

12. L. N. Bhuyan and D. P. Agrawal, "Design and performance of generalized interconnection networks," IEEE Transactions on Computers, Vol. C-32, No. 12, December 1983, pp. 1081-1089.

13. L. N. Bhuyan and C. W. Lee, "An interference analysis of interconnection networks," Proceedings of the 1983 International Conference on Parallel Processing, pp. 2-9.

14. F. A. Briggs and E. S. Davidson, "Organization of semiconductor memories for parallel-pipelined processors," IEEE Transactions on Computers, Vol. C-26, No. 2, February 1977, pp. 162-169.

15. F. A. Briggs and M. Dubois, "Cache effectiveness in multiprocessor systems with pipelined parallel memories," Proceedings of the 1981 International Conference on Parallel Processing, Columbus, OH, August 25-28, 1981, pp. 306-313.

16. F. A. Briggs and M. Dubois, "Effectiveness of private caches in multiprocessor systems with parallel-pipelined memories," IEEE Transactions on Computers, Vol. C-32, No. 1, January 1983, pp. 48-59.

17. G. Broomell and J. R. Heath, "Classification categories and historical development of circuit switching topologies," ACM Computing Surveys, Vol. 15, No. 2, June 1983, pp. 95-133.

18. D. Y. Chang, D. J. Kuck, and D. H. Lawrie, "On the effective bandwidth of parallel memories," IEEE Transactions on Computers, Vol. C-26, No. 5, May 1977, pp. 480-490.

19. P.-Y. Chen, D. H. Lawrie, P.-C. Yew, and D. A. Padua, "Interconnection networks using shuffles," Computer, Vol. 14, No. 12, December 1981, pp. 55-64.

20. P.-Y. Chen, P.-C. Yew, and D. Lawrie, "Performance of packet switching in buffered single-stage shuffle-exchange networks," Proceedings of the 1982 International Conference on Distributed Computing Systems, IEEE, pp. 622-627.

21. V. Cherkassky and M. Malek, "On permuting properties of regular rectangular SW-banyans," IEEE Transactions on Computers, Vol. C-34, No. 6, June 1985, pp. 542-546.

22. V. S. Cherkassky, "Performance of non-rectangular multistage interconnection networks," Proceedings of the 6th International Conference on Distributed Computing Systems, Cambridge, MA, May 19-23, 1986, IEEE, pp. 2-7.

23. C.-Y. Chin and K. Hwang, "Connection principles for multipath packet switching networks," 11th Annual International Symposium on Computer Architecture: Conference Proceedings, Ann Arbor, MI, June 5-7, 1984, pp. 99-108.

24. C.-Y. Chin and K. Hwang, "Packet switching networks for multiprocessors and data flow computers," IEEE Transactions on Computers, Vol. C-33, No. 11, November 1984, pp. 991-1003.

25. G. Chiola, M. A. Marsan, and G. Balbo, "Product-form solution techniques for the performance analysis of multiple-bus multiprocessor systems with nonuniform memory references," IEEE Transactions on Computers, Vol. 37, No. 5, May 1988, pp. 532-540.

26. R. Contemo and R. Melen, "An analytical model for a class of processor-memory interconnection networks," IEEE Transactions on Computers, Vol. C-36, No. 11, November 1987, pp. 1374-1378.

27. Z. Cvetanovic, "Best and worst mappings for the omega network," IBM Journal of Research and Development, Vol. 31, No. 4, July 1987, pp. 452-463.

28. C. R. Das and L. N. Bhuyan, "Computation availability of multiple-bus multiprocessors," Proceedings of the 1985 International Conference on Parallel Processing, St. Charles, IL, August 20-23, 1985, pp. 807-813.

29. C. R. Das and L. N. Bhuyan, "Bandwidth availability of multiple-bus multiprocessors," IEEE Transactions on Computers, Vol. C-34, No. 10, October 1985, pp. 918-926.

30. N. J. Davis IV and H. J. Siegel, "The performance analysis of partitioned circuit switched multistage interconnection networks," 12th Annual International Symposium on Computer Architecture: Conference Proceedings, Boston, MA, June 17-19, 1985, pp. 387-394.

31. N. J. Davis IV and H. J. Siegel, "Performance studies of multiple-packet multistage cube networks and comparison to circuit switching," Proceedings of the 1986 International Conference on Parallel Processing, St. Charles, IL, August 20-23, 1986, pp. 108-114.

32. D. M. Dias and J. R. Jump, "Analysis and simulation of buffered delta networks," IEEE Transactions on Computers, Vol. C-30, No. 4, April 1981, pp. 273-282.

33. D. M. Dias and J. R. Jump, "Packet switching interconnection networks for modular systems," Computer, Vol. 14, No. 12, December 1981, pp. 43-53.

34. H.-C. Du, "On the performance of synchronous multiprocessors," IEEE Transactions on Computers, Vol. C-34, No. 5, May 1985, pp. 462-466.

35. M. Dubois, "Effect of invalidations on the hit ratio of cache-based multiprocessors," Proceedings of the 1987 International Conference on Parallel Processing, St. Charles, IL, August 17-21, 1987, pp. 255-257.

36. M. Dubois, "Throughput analysis of cache-based multiprocessors with multiple buses," IEEE Transactions on Computers, Vol. 37, No. 1, January 1988, pp. 58-70.

37. M. Dubois and F. A. Briggs, "Efficient interprocessor communication for MIMD systems," 8th Annual International Symposium on Computer Architecture: Conference Proceedings, 1981, pp. 187-196.

38. M. Dubois and F. A. Briggs, "Performance of synchronized iterative processes in multiprocessor systems," IEEE Transactions on Software Engineering, Vol. SE-8, No. 4, July 1982, pp. 419-431.

39. T.-y. Feng, "A survey of interconnection networks," Computer, Vol. 14, No. 12, December 1981, pp. 12-27.

40. M. A. Franklin, "VLSI performance comparison of banyan and crossbar communications networks," IEEE Transactions on Computers, Vol. C-30, No. 4, April 1981, pp. 283-290.

41. A. Fukuda, "Equilibrium point analysis of memory interference in multiprocessor systems," Denshi Tsushin Gakkai Ronbunshi, Vol. 68-D, No. 8, August 1985, pp. 1441-1448. English translation in Systems and Computers in Japan, Vol. 17, No. 4, April 1986, pp. 84-93. Revised version in IEEE Transactions on Computers, Vol. 37, No. 5, May 1988, pp. 585-593.

42. O. Ganz and I. Chlamtac, "Analysis of multiple-bus synchronous and asynchronous communication systems," Performance Evaluation, Vol. 5, No. 1, February 1985, pp. 1-13.

43. U. Garg and Y.-P. Huang, "Decomposing banyan networks for performance analysis," IEEE Transactions on Computers, Vol. 37, No. 3, March 1988, pp. 371-376.

44. I. Gazit and M. Malek, "On the number of permutations performable by extra-stage multistage interconnection networks," Proceedings of the 1987 International Conference on Parallel Processing, St. Charles, IL, August 17-21, 1987, pp. 461-470.

45. L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," 1st Annual International Symposium on Computer Architecture: Conference Proceedings, 1973, pp. 21-28.

46. A. Gottlieb and J. T. Schwartz, "Networks and algorithms for very-large scale parallel computation," Computer, Vol. 15, No. 1, January 1982, pp. 27-36.

47. P. A. Grasso, T. S. Dillon, and K. E. Forward, "Memory interference in multimicroprocessor systems with a time-shared bus," IEE Proceedings, Vol. 131, Pt. E, No. 2, March 1984, pp. 61-68.

48. P. A. Grasso, T. S. Dillon, and K. E. Forward, "Performance analysis of common bus multimicroprocessor systems," presented at the International Workshop on Modeling and Performance Evaluation of Parallel Systems, Grenoble, France, December 13-18, 1984. In Journal of Systems and Software, Vol. 6, No. 1&2, May 1986, pp. 71-79.

49. M. A. Holliday and M. K. Vernon, "Exact performance estimates for multiprocessor memory and bus interference," IEEE Transactions on Computers, Vol. C-36, No. 1, January 1987, pp. 76-85.

50. C. H. Hoogendoorn, "A general model for memory interference in multiprocessors," IEEE Transactions on Computers, Vol. C-26, No. 10, October 1977, pp. 998-1005.

51. K. Hwang and F. A. Briggs, Computer Architecture and Parallel Processing. New York: McGraw-Hill, 1984.

52. K. B. Irani and I. H. Onyuksel, "A closed-form solution for the performance analysis of multiple-bus multiprocessor systems," IEEE Transactions on Computers, Vol. C-33, No. 11, November 1984, pp. 1004-1012.

53. Y. A. Kogan and L. B. Boguslavsky, "Asymptotic analysis of memory interference in multiprocessors with private cache memories," Performance Evaluation, Vol. 5, No. 2, May 1985, pp. 97-104.

54. Y. A. Kogan, A. I. Lyakhov, and S. G. Nersesyan, "Asymptotic analysis of cache memory performance in multiprocessor systems," Avtomatika i Telemekhanika, No. 11, November 1986, pp. 142-151. English translation in Automation and Remote Control, Vol. 47, No. 11, Part 2, November 1986, pp. 1584-1593.

55. J. Kriz, "A queueing analysis of a symmetric multiprocessor with shared memories and buses," IEE Proceedings, Vol. 130, Pt. E, No. 3, May 1983, pp. 83-89.

56. C. P. Kruskal and M. Snir, "The performance of multistage interconnection networks for multiprocessors," IEEE Transactions on Computers, Vol. C-32, No. 12, December 1983, pp. 1091-1098.

57. C. P. Kruskal, M. Snir, and A. Weiss, "The distribution of waiting times in clocked multistage interconnection networks," Proceedings of the 1986 International Conference on Parallel Processing, St. Charles, IL, August 19-22, 1986, pp. 12-19.

58. M. Kumar, D. M. Dias, and J. R. Jump, "Switching strategies in a class of packet switching networks," 10th Annual International Symposium on Computer Architecture: Conference Proceedings, June 13-17, 1983, pp. 284-300.

59. M. Kumar and J. R. Jump, "Performance of unbuffered shuffle-exchange networks," IEEE Transactions on Computers, Vol. C-35, No. 6, June 1986, pp. 573-578.

60. V. P. Kumar and S. M. Reddy, "Augmented shuffle-exchange multistage interconnection networks," Computer, Vol. 20, No. 6, June 1987, pp. 30-40.

61. T. Lang, "Interconnections between processors and memory modules using the shuffle-exchange network," IEEE Transactions on Computers, Vol. C-25, No. 5, May 1976, pp. 496-503.

62. T. Lang and H. S. Stone, "A shuffle-exchange network with simplified control," IEEE Transactions on Computers, Vol. C-25, No. 1, January 1976, pp. 55-65.

63. 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, December 1982, pp. 1227-1234.

64. D. H. Lawrie, "Access and alignment of data in an array processor," IEEE Transactions on Computers, Vol. C-24, No. 12, December 1975, pp. 1145-1155.

65. K. Y. Lee and D. Lee, "On the augmented data manipulator network in SIMD environments," IEEE Transactions on Computers, Vol. 37, No. 5, May 1988, pp. 574-584.

66. M. Lee and C.-L. Wu, "Performance analysis of circuit switching baseline interconnection networks," 11th Annual International Symposium on Computer Architecture: Conference Proceedings, Ann Arbor, MI, June 5-7, 1984, pp. 82-90.

67. R. L. Lee, P.-C. Yew, and D. H. Lawrie, "Multiprocessor cache design considerations," 14th Annual International Symposium on Computer Architecture: Conference Proceedings, Pittsburgh, PA, June 2-5, 1987, pp. 253-262.

68. Y.-C. Liu and C.-J. Jou, "Effective memory bandwidth and processor blocking probability in multiple-bus systems," IEEE Transactions on Computers, Vol. C-36, No. 6, June 1987, pp. 761-764.

69. P. Markenscoff, "Markov models for a multiple processor system with a shared bus," IEE Proceedings, Vol. 132, Pt. E, No. 6, November 1985, pp. 316-322.

70. M. A. Marsan, G. Balbo, G. Chiola, and S. Donatelli, "On the product-form solution of a class of multiple-bus multiprocessor system models," presented at the International Workshop on Modeling and Performance Evaluation of Parallel Systems, Grenoble, France, December 13-18, 1984. In Journal of Systems and Software, Vol. 6, No. 1&2, May 1986, pp. 117-124.

71. M. A. Marsan, G. Balbo, and G. Conte, "Comparative performance analysis of single bus multiprocessor architectures," IEEE Transactions on Computers, Vol. C-31, No. 12, December 1982, pp. 1179-1191.

72. M. A. Marsan, G. Balbo, G. Conte and F. Gregoretti, "Modeling bus contention and memory interference in a multiprocessor system," IEEE Transactions on Computers, Vol. C-32, No. 1, January 1983, pp. 60-72.

73. M. A. Marsan and M. Gerla, "Markov models for multiple bus multiprocessor systems," IEEE transactions on Computers, Vol. C-31, No. 3, March 1982, pp. 239-248.

74. T. N. Mudge and H. B. Al-Sadoun, "Memory interference models with variable connection time," IEEE Transactions on Computers, Vol. C-33, No. 11, November 1984, pp. 1033-1038.

75. T. N. Mudge and H. B. Al-Sadoun, "A semi-Markov model for the performance of multiple-bus systems," Proceedings of the 1985 International Conference on Parallel Processing, St. Charles, IL, August 20-23, 1985, pp. 521-530.

76. T. N. Mudge, H. B. Al-Sadoun, and B. A. Makrucki, "Memory-interference model for multiprocessors based on semi-Markov processes," IEE Proceedings, Vol. 134, Pt. E, No. 4, July 1987, pp. 203-214.

77. T. N. Mudge, J. P. Hayes, G. D. Buzzard, and D. C. Winsor, "Analysis of multiple bus interconnection networks," Proceedings of the 1984 International Conference on Parallel Processing, pp. 228-232.

78. T. N. Mudge, J. P. Hayes, G. D. Buzzard, and D. C. Winsor, "Analysis of of multiple-bus interconnection networks," Journal of Parallel and Distributed Computing, Vol. 3, No. 3, September 1986, pp. 328-343.

79. T. N. Mudge, J. P. Hayes, and D. C. Winsor, "Multiple bus architectures," Computer, Vol. 20, No. 6, June 1987, pp. 42-48.

80. T. N. Mudge and B. A. Makrucki, "Probabilistic analysis of a crossbar switch," 9th Annual International Symposium on Computer Architecture: Conference Proceedings, 1982, pp. 311-320.

81. A. Norton and G. F. Pfister, "A methodology for predicting multiprocessor performance," Proceedings of the 1985 International Conference on Parallel Processing, St. Charles, IL, August 20-23, 1985, pp. 772-781.

82. K. Padmanabhan and D. H. Lawrie, "Performance analysis of redundant-path networks for multiprocessor systems," ACM Transactions on Computer Systems, Vol. 3, No. 2, May 1985, pp. 117-144.

83. D. S. Parker, Jr., "Notes on shuffle/exchange - type switching networks," IEEE Transactions on Computers, Vol. C-29, No. 3, March 1980, pp. 213-222.

84. D. S. Parker and C. S. Raghavendra, "The Gamma network," IEEE Transactions on Computers, Vol. C-33, No. 4, April 1984, pp. 367-373.

85. J. H. Patel, "Processor-memory interconnections for multiprocessors," 6th Annual International Symposium on Computer Architecture: Conference Proceedings, Philadelphia, PA, April 23-25, 1979, pp. 168-177.

86. J. H. Patel, "Performance of processor-memory interconnections for multiprocessors," IEEE Transactions on Computers, Vol. C-30, No. 10, October 1981, pp. 771-780.

87. J. H. Patel, "A performance model for multiprocessors with private cache memories," Proceedings of the 1981 International Conference on Parallel Processing, Columbus, OH, August 25-28, 1981, pp. 314-317.

88. J. H. Patel, "Analysis of multiprocessors with private cache memories," IEEE Transactions on Computers, Vol. C-31, No. 4, April 1982, pp. 296-304.

89. M. C. Pease III, "The indirect binary n-cube microprocessor array," IEEE Transactions on Computers, Vol. C-26, No. 5, May 1977, pp. 458-473.

90. A. Pombortsis and C. Halatsis, "Performance of crossbar interconnection networks in presence of 'hot spots'," Electronics Letters, Vol. 24, No. 3, February 4, 1988, pp. 182-184.

91. B. R. Rau, "Program behavior and the performance of interleaved memories," IEEE Transactions on Computers, Vol. C-28, No. 3, March 1979, pp. 191-199.

92. B. R. Rau, "Interleaved memory bandwidth in a model of a multiprocessor computer system," IEEE Transactions on Computers, Vol. C-28, No. 9, September 1979, pp. 678-681.

93. C. V. Ravi, "On the bandwidth and interference in interleaved memory systems," IEEE Transactions on Computers, Vol. C-21, No. 8, August 1972, pp. 899-901.

94. K. V. Sastry and R. Y. Kain, "On the performance of certain multiprocessor computer organizations," IEEE Transactions on Computers, Vol. C-24, No. 11, November 1975, pp. 1066-1074.

95. A. A. Sawchuk, B. K. Jenkins, C. S. Raghavendra, and A. Varma, "Optical crossbar networks," Computer, Vol. 20, No. 6, June 1987, pp. 50-62.

96. A. S. Sethi and N. Deo, "Interference in multiprocessor systems with localized memory access probabilities," IEEE Transactions on Computers, Vol. C-28, No. 2, February 1979, pp. 157-163.

97. H. J. Siegel, Interconnection Networks for Large-Scale Parallel Processing. Lexington, MA: Lexington Books, 1985.

98. H. J. Siegel and R. J. McMillen, "Using the augmented data manipulator network in PASM," Computer, Vol. 14, No. 2, February 1981, pp. 25-33.

99. H. J. Siegel and R. J. McMillen, "The multistage cube: a versatile interconnection network," Computer, Vol. 14, No. 12, December 1981, pp. 65-76.

100. H. J. Siegel and S. D. Smith, "Study of multistage SIMD interconnection networks," 5th Annual International Symposium on Computer Architecture: Conference Proceedings, 1978, pp. 223-229.

101. K. O. Siomalas and B. A. Bowen, "Performance of cross-bar multiprocessor systems," IEEE Transactions on Computers, Vol. C-32, No. 7, July 1983, pp. 689-695.

102. B. Smilauer, "General model for memory interference in multiprocessors and mean value analysis," IEEE Transactions on Computers, Vol. C-34, No. 8, August 1985, pp. 744-751.

103. T. H. Szymanski, "On the universality of multistage interconnection networks," Proceedings of the 1986 International Conference on Parallel Processing, St. Charles, IL, August 19-22, 1986, pp. 316-323.

104. T. H. Szymanski and V. C. Hamacher, "On the permutation capability of multistage interconnection networks," IEEE Transactions on Computers, Vol. C-36, No. 7, July 1987, pp. 810-822.

105. S. Thanawastien and V. P. Nelson, "Interference analysis of shuffle/exchange networks," IEEE Transactions on Computers, Vol. C-30, No. 8, August 1981, pp. 545-556.

106. S. Thanawastien and P. K. Srimani, "The universality of a class of modified single-stage shuffle-exchange networks," IEEE Transactions on Computers, Vol. 37, No. 3, March 1988, pp. 348-352.

107. K. J. Thurber, "Circuit switching technology: a state-of-the-art survey," Digest of Papers: COMPCON, Fall 1978, IEEE, pp. 116-124.

108. D. Towsley, "Approximate models of multiple bus multiprocessor systems," IEEE Transactions on Computers, Vol. C-35, No. 3, March 1986, pp. 220-228.

109. M. Valero, E. Sanvicente, J. M. Llaberia, T. Lang, and J. Labarta, "A performance evaluation of the multiple bus network for multiprocessor systems," Proceedings of the 1983 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Minneapolis, MN, August 29-31, 1983, pp. 200-206.

110. A. Varma and C. S. Raghavendra, "Performance analysis of a redundant-path interconnection network," Proceedings of the 1985 International Conference on Parallel Processing, St. Charles, IL, August 20-23, 1985, pp. 474-479.

111. M. Witten, B. L. Bodnar and A. C. Liu, "Simulation and modeling of a single bus tightly coupled multiprocessor system," Mathematics and Computers in Simulation, Vol. 29, No. 1, February 1987, pp. 19-31.

112. C.-L. Wu and T.-Y. Feng, "Routing techniques for a class of multistage interconnection networks," Proceedings of the 1978 International Conference on Parallel Processing, pp. 197-205.

113. C.-L. Wu and T.-Y. Feng, "On a class of multistage interconnection networks," IEEE Transactions on Computers, Vol. C-29, No. 8, August 1980, pp. 694-702.

114. C.-L. Wu and T.-Y. Feng, "The universality of the shuffle-exchange network," IEEE Transactions on Computers, Vol. C-30, No. 5, May 1981, pp. 324-332.

115. Q. Yang and S. G. Zaky, "Communication performance in multiple-bus systems," IEEE Transactions on Computers, Vol. 37, No. 7, July 1988, pp. 848-853.

116. P. C. C. Yeh, J. H. Patel, and E. S. Davidson, "Shared cache for multiple-stream computer systems," IEEE Transactions on Computers, Vol. C-32, No. 1, January 1983, pp. 38-47.

117. P. C. C. Yeh, J. H. Patel, and E. S. Davidson, "Performance of shared cache for parallel-pipelined computer systems," 10th Annual International Symposium on Computer Architecture: Conference Proceedings, 1983, pp. 117-123.

118. D. W. L. Yen, J. H. Patel, and E. S. Davidson, "Memory interference in synchronous multiprocessor systems," IEEE Transactions on Computers, Vol. C-31, No. 11, November 1982, pp. 1116-1121.

119. H. Yoon, K. Y. Lee, and M. T. Liu, "Performance analysis and comparison of packet switching interconnection networks," Proceedings of the 1987 International Conference on Parallel Processing, St. Charles, IL, August 17-21, 1987, pp. 542-545.

-114A (REV. 2-80)

U.S. DEPT. OF COMM.

BIBLIOGRAPHIC DATA SHEET *(See instructions)*

1. PUBLICATION OR REPORT NO.
NISTIR 88-3857

2. Performing Organ. Report No.

3. Publication Date
SEPTEMBER 1988

TITLE AND SUBTITLE

Performance Evaluation of Shared-Memory Computers

AUTHOR(S)

James R. Nechvatal

PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

PONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

Computer Systems Technology Center
)ffice of Export Administration
)epartment of Commerce
Herbert C. Hoover Building
.4th and Constitution Avenue, N.W.
Washington, DC 20230

SUPPLEMENTARY NOTES

□ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

BSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant ibliography or literature survey, mention it here)*

e survey and discuss analytic performance models for shared-memory computers. ır purpose is to assess these models to determine if they are relevant to the ısk of assisting the Office of Export Administration in establishing guidelines ır exports of computers. The focus of this study is on the assumptions made by delers regarding interconnection networks and memory access patterns exhibited applications. The great majority of models concentrate on analyzing a small llection of networks. Furthermore, most modelers make strong assumptions garding the independence and uniform distribution of memory accesses. The nsequences of these assumptions in the resulting models are noted.

Y WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

nputers; export; memory; models; networks; performance

AILABILITY

Unlimited

For Official Distribution. Do Not Release to NTIS

Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES
29

15. Price
$11.95