PERGAMON

Computers and Electrical Engineering 28 (2002) 43–57

Computers and Electrical Engineering

www.elsevier.com/locate/compeleceng

A comparative study of bidirectional ring and crossbar interconnection networks

Hitoshi Oi [a,*], N. Ranganathan [b]

[a] *Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA*
[b] *Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA*

Received 16 June 1999; accepted 3 April 2000

Abstract

For distributed shared memory multiprocessors, the choice and the design of interconnection networks have a significant impact on their performance. Bidirectional ring networks are considered to be physically fast due to their simple structure, but topologically slow since their communication latency grows proportionally to the number of nodes. In this paper, we will present a quantitative measure to the question of how physically fast a bidirectional ring has to be to overcome its topologically slow communication latency by comparing it to the crossbar, an interconnection network that has opposite characteristics to the ring network.

A hybrid method, in which workload parameters extracted from memory traces are given to an analytical model is used for performance evaluation. Our study shows for 16 nodes configuration, the performance of two networks are similar. For 32 and 64 nodes configurations, the bidirectional ring outperforms the crossbar by 21% and 61% respectively, on the average of four parallel applications. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Interconnection networks; Distributed shared memory multiprocessor; Performance evaluation; Slotted ring; Crossbar

1. Introduction

A distributed shared memory (DSM) multiprocessor provides a view of a globally shared address space by sending coherence protocol messages between processing elements. Therefore, the interconnection network affects the performance of a DSM multiprocessor significantly.

* Corresponding author. Tel.: +1-408-341-5546.
E-mail addresses: hitoshi@cse.fau.edu (H. Oi), ranganat@csee.usf.edu (N. Ranganathan).

0045-7906/01/$ - see front matter © 2001 Elsevier Science Ltd. All rights reserved.
PII: S0045-7906(00)00044-6

Ring networks have the advantages of (1) fixed node degree (modular expandability), (2) simple network interface structure (fast operation speed) and (3) low wiring complexity (fast transmission speed). On the other hand, the main disadvantage of the ring network is its communication latency growth rate is proportional to the number of nodes. However, when non-local misses are likely to be destined to neighboring nodes, this disadvantage can be alleviated by using a bidirectional ring [1,2]. Crossbar switches have opposite characteristics: they can connect any pair of nodes by one hop, but the transmission speed is expected to be much slower due to wiring and circuit complexity [3]. In this paper, we will present a quantitative measure to the question of how physically fast a bidirectional ring has to be to overcome its topologically slow communication latency, by comparing its performance to that of the crossbar switch.

This paper is organized as follows: The rest of this section introduces the past studies on the comparisons of the interconnection networks for multiprocessors and instances of multiprocessor using either ring or crossbar switch networks. The architectures of multiprocessors that will be assumed in our study are presented in Section 2. Our methodology for evaluating the performance and the profile of the applications used in the performance evaluation are described in Section 3. The comparison of the bidirectional ring and the crossbar by estimated execution times (ETs) is presented in Section 4. Some conclusions are provided in Section 5.

1.1. Related work

There have been several researches in the comparisons of interconnection networks for multiprocessors in the past. Ravindran and Stumn compared the performance of multiprocessors using hierarchical ring and mesh networks [4]. The miss latency was used for performance comparison. Their study mainly showed maximum number of nodes at which the hierarchical ring outperformed the mesh network. Since they assumed a unidirectional ring, nearest-neighbor communication pattern was not taken into account. Barroso and Dubois evaluated the performance of the slotted ring multiprocessor [5]. They investigated the effect of the design choices such as coherence protocols (snoopy, linked list and full-map directory) and processor speed, and compared their unidirectional ring architecture with the split-transaction bus. Lang et al. studied the effective bandwidth of the crossbar switches, and compared it to that of the multiple-bus interconnection network using parametric simulations [3]. They assumed the dance-hall UMA architecture (processors and memories are different sides of interconnection network).

The instances of multiprocessors using crossbar switches or ring networks include the following. NUMAchine [6] is a multiprocessor developed at the University of Toronto. It has three hierarchy levels: a small number of PEs connected by a bus form a cluster (called "station") and the stations are connected by two levels of (local and central) rings. KSR-1 of Kendall Square Research was a cache-only memory architecture (COMA) multiprocessor with hierarchy rings [7]. Exemplar S-Class of Hewlett Packard [8] and CP-PACS developed at University of Tsukuba [9] use crossbar switches. Convex SPP-1000 [10] uses a crossbar switch within a cluster and scalable coherence interface (SCI) [11] rings between clusters.

2. Architecture model

In this section, we describe the architecture of the DSM multiprocessors using the bidirectional ring and the crossbar that will be assumed in this paper. Each processing element (PE) consists of a processor, a memory unit with directory entries, L1 and L2 cache, and a network interface (Fig. 1). PEs are connected by either a bidirectional ring or a crossbar switch network.

It is assumed that both architectures are based on the CC-NUMA model with directory based cache coherency protocol. The memory unit within a PE is a part of the globally shared memory and is associated with directory entries corresponding to the part of global address space assigned to the PE. L1 cache is direct mapped, write-through and its access time and size are one clock cycle and 16 KB. L2 cache is full associative, write-back and its access time (t_{L2}) and size are five clock cycle and infinity. Block size of both L1 and L2 is 16 bytes. L2 cache miss is responded either by the home node (the PE that is assigned the portion of global address of the accessed data) or by the owner node (the PE that owns a modified copy of the requested data in the cache).

For a write access, invalidation scheme is used. On the bidirectional ring, a single invalidation message is broadcast by passing it all the way through the ring. On the other hand, the crossbar network is assumed to have the multicast functionality to send invalidate messages simultaneously to all the PEs that have copies of the block.

The parameters that define the configuration of the system are listed in Table 1. t_{L2}, t_s, t_r, t_m, and t_c are no-contention timing parameters, and they are represented in terms of processor clock cycles. In this paper, we consider small to medium scale multiprocessors, and hence we chose $N = 16$, 32 and 64. The speed of crossbar network is fixed, while that of ring network is varied widely to find out the point where the performance of two configurations match.

Both bidirectional ring and crossbar are assumed to be flat (one level). On the bidirectional ring, a header message is divided into two packets, and it takes t_r to transfer a packet between

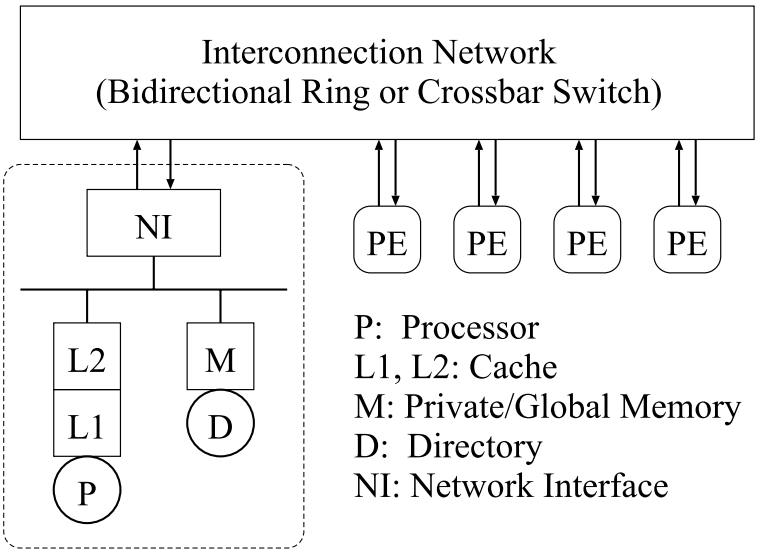


Fig. 1. DSM multiprocessor architecture.

Table 1
System parameters (timing parameters are in processor clock cycles)

Symbol	Description	Value
N	Number of PE	16, 32, 64
t_{L2}	L2 cache access time	5
t_s	Crossbar switch packet transfer time	4, 8, 16, 32, 64
t_r	Bidirectional ring packet transfer time	1–5
t_m	Memory access time	50
t_c	Protocol handling time	10
d_p	Data message packet length	4

adjacent PEs. This assumption of t_r is to compensate that a node on the bidirectional ring has two links for each direction. On the bidirectional ring, each link connects a pair of adjacent PEs with a minimum wire length. Therefore, it is not difficult to keep t_r constant for larger N. Thus, we use the same $t_r = 1, \ldots, 5$ for all N.

t_s is the time to transfer a header message between any pair of PEs on the crossbar network. We have chosen t_s as follows: First, we take two instances of shared memory multiprocessors using crossbar switches, and look at their relative speed between the processor and the crossbar, the dimensional size and the data path width of the crossbar, and the word length of the processor. Exemplar S-Class of Hewlett Packard uses 180 MHz 64-bits processors and 8×8 crossbars operating at 120 MHz with 64-bits datapath [8]. CP-PACS developed at University of Tsukuba uses 180 MHz 64-bits processors and 17×17 crossbar switches with data bandwidth of 300 MB/s per node [9]. Using the above definition, Exemplar and CP-PACS have $t_s = 1.5$ and $t_s = 8$ respectively. Taking these values into consideration, we have chosen $t_s = 4$ and 8 for $N = 16$. Next, we consider t_s for larger N (32 and 64). The delay of crossbar switch grows at $O(N^2)$ due to the length of wire and number of nodes [3]. Thus, we use $t_s = 16$ and $t_s = 64$ for $N = 32$ and 64, respectively. Unlike UMA (including Exemplar) in which the delay of crossbar is dominant for the network latency, both the wire connection between PEs and crossbar, which is considered to be $O(N)$, and the delay of crossbar switches affect the latency of the network in NUMA architecture. Hence, we also use lower values of $t_s = 8$ and $t_s = 16$ and 32 for $N = 32$ and 64, respectively.

A data message is assumed to be four times longer than a header message ($d_p = 4$). This value is chosen from the cache block size (16 bytes for 32 bit processors) assumed when the trace files were collected. On the crossbar, a data message is transmitted in a contiguous $4t_s$ time period, while on the bidirectional ring a data message is divided into multiple packets and they are sent in (possibly) uncontiguous slots. Therefore, the transmission delay of a data message on the bidirectional ring is $\geqslant 8t_r$. [1] We assume the slotted ring configuration for the bidirectional ring.

3. Performance model

The methods that have been used for performance evaluation of computer systems include analytical models [12], parametric simulations [4], trace-driven simulations [13] and execution-

[1] Equality holds when all the packets are transmitted in contiguous slots.

driven simulations [14]. In this paper, we use the hybrid approach by Barroso and Dubois [5] by extending it to the bidirectional ring and the crossbar. Below, we briefly describe the derivation of *ET* using the hybrid approach.

The *ET* in clock cycles is given by

$$ET = Inst + DataAccess, \tag{1}$$

where *Inst* is instruction fetches. We assume that all the instruction fetches are L1 cache hit. Let *Refs*, *L1Miss*, *L2Miss*, and *L2MissLatency* be the number of references, the number of L1 cache misses, the number of L2 cache misses, and the penalty of L2 cache miss, respectively. Thus, Eq. (1) becomes

$$ET = Refs + L1Miss \times t_{L2} + L2Miss \times L2MissLatency. \tag{2}$$

L2 cache miss is responded by either local or remote memory. Thus, L2MissLatency is defined to consist of T, the time to transmit a message (either request or data) to the network, P, the propagation delay of a message to reach the destination on the network, M, the time to access the memory, and C, the time of coherence protocol handling. These components of miss latency appear different number of times per miss depending on the access mode and the state of the memory block. Note that M includes contention delay while t_m in Table 1 does not.

P is constant t_s on the crossbar while it is the average message traversal length extracted from the traces on the bidirectional ring. C is assumed to be constant t_c (i. e. no contention at L2 cache is modeled).

We model the contention at network and memory. Thus, the values of parameters T and M are the functions of the utilizations of the resources, network and memory, respectively. The utilization of the bidirectional ring R_u is

$$R_u = \frac{t_r(\sum H\,dTrv + d_p \sum DtTrv)}{ET},$$

where $\sum H\,dTrv$ and $\sum DtTrv$ are the sums of header and data message traversals, respectively. On the other hand, the utilization of the crossbar switch is

$$SW_u = \frac{t_s(\sum H\,dTx + d_p \sum H\,dTx)}{ET},$$

where $\sum H\,dTx$ and $\sum DtTx$ are the the numbers of header and data message transmissions. This is because on the crossbar switch network, a message only takes one hop to reach any destination PE. These utilization parameters are combined with the M/G/1 queue model to estimate the ET of applications on both architectures [15].

Thus, ET itself is a function of ET.

$$ET = Refs + L1Miss \times t_{L2} + L2Miss \times L2MissLatency(ET).$$

The derivation of *ET* is iterated until it converges to a tolerant level (<0.1%). We use two metrics to evaluate the performance of two architectures. One is the ratio of *ETs*, namely, the *ET* on the crossbar divided by the *ET* on the bidirectional ring. Another metric is the scalability (speed up), which is obtained by dividing the *ET* of the uniprocessor by that of the multiprocessor.

Table 2
Profile of trace files

Application	Refs (Inst)	N	Miss Rate (%)			
			L1	L2	Write	Local
FFT	7.44M	16	23.94	5.08	77.98	52.46
	(3.11M)	32	24.84	5.99	66.30	50.87
		64	25.78	6.92	57.41	50.09
Simple	27.03M	16	19.78	4.69	31.14	24.02
	(11.59M)	32	20.28	5.32	28.04	20.56
		64	21.84	7.10	22.35	11.41
Weather	31.76M	16	9.18	2.41	59.20	42.54
	(13.64M)	32	9.33	2.61	56.99	39.66
		64	10.33	3.56	42.57	23.90
Speech	22.55M	16	13.80	12.88	84.29	67.20
	(11.77M)	32	14.08	13.54	81.96	66.14
		64	14.64	14.30	78.58	65.60

3.1. Trace files

The profile of the memory traces we use in our experiments are listed in Table 2. These trace files are from the MIT trace set [16], and are obtained from the ftp server of the TraceBase project at the New Mexico State University [17]. FFT, Simple and Weather were written in FORTRAN, and traces of these applications were derived using the postmortem method, a technique that generates a parallel trace from a uniprocessor execution. Speech was written in the programming language Mul-T, and its trace file was collected by inserting instrumentation codes into the program by the compiler. Write miss rate (sixth column) is the fraction of write misses within all L2 cache misses. A write access to shared block is considered to be a write miss since it generates memory access and network traffic.

The workload parameters of each application were extracted from the trace file as follows: Each trace file is fed to a cache/directory simulator developed at the laboratory for computer science of MIT, [2] and statistical information of the applications such as hit/miss, read/write, clean/dirty was collected. These trace files were collected on a 64 processor system. To extract data for the case of 16 (32) processors, access traces from $4i$ to $4i + 3$ (from $2i$ to $2i + 1$) were given to ith cache, where $i = 0, \ldots, 15$ $(i = 0, \ldots, 31)$. It is assumed that all the instruction fetches are cache hits.

In addition to the original functionality, two modifications were made on the cache simulator. First, two-level cache hierarchy was incorporated as described in the previous section. Another modification made to the cache simulator was to collection of communication distance (number of links on the bidirectional ring) between the node where miss occurs and the home node. We assume that the first PE that accesses a block is the home node of the block. When the miss node and the home node are the same, no network traffic is generated. On FFT and Speech, more than

[2] This simulator was also provided by the TraceBase ftp server.

half of accesses to shared data are to local addresses, while on Simple and Weather the locality of shared access is quite low. This communication distance affects the performance of the bidirectional ring significantly in two ways: First, the longer the distance the longer the propagation delay (higher P). Second, the longer the distance the higher utilization of the ring (higher T). Figs. 12–14 show the distribution of request – home node distance on the bidirectional ring. For $N = 64$, the difference of communication pattern among applications is observed most clearly.

FFT and Speech exhibit all-to-all communication pattern while Weather and Simple have a certain level of nearest neighbor communication patterns. Especially for $N = 64$ on Weather, 30% of non-local misses are accesses to adjacent nodes. For $N = 16$ and 32, the degree of nearest neighbor communication pattern of Weather and Simple are weaker than the case of $N = 64$. Although the number of applications is not large, these applications exhibit variations in their workload parameters.

4. Bidirectional ring versus crossbar evaluation

In this section, we present comparisons of the bidirectional ring and the crossbar using the model and the traces presented in the previous section (Figures in this section are placed after the references).

FFT (Fig. 2) has all-to-all communication pattern (Fig. 14), which is an advantage to the crossbar over the bidirectional ring. The speed up for larger N is medium among four applications (Fig. 3). L2 miss rate of FFT is relatively small, and it increases by 36% for $N = 16 \rightarrow 64$. However, its local miss rate is almost constant over N. This suppresses the increase of network traffic, and is advantageous for both architecture. Write miss rate of FFT decreases from 78% to 57% for $N = 16 \rightarrow 64$. This is an advantage for the bidirectional ring since on the bidirectional

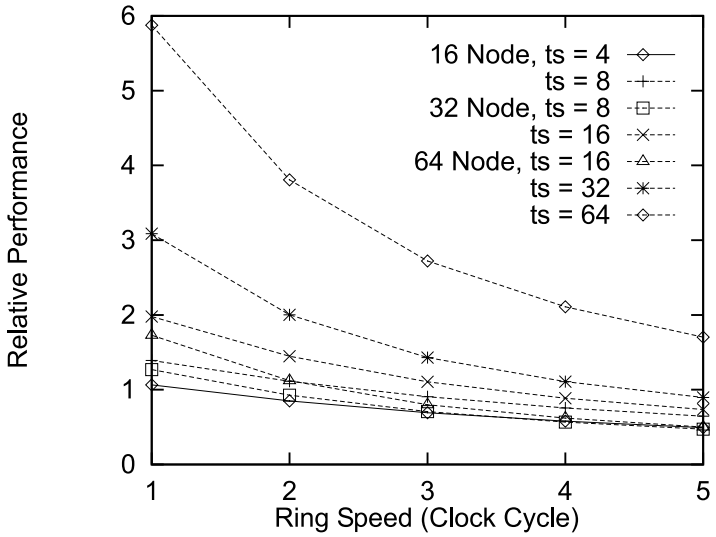


Fig. 2. Relative performance/FFT.

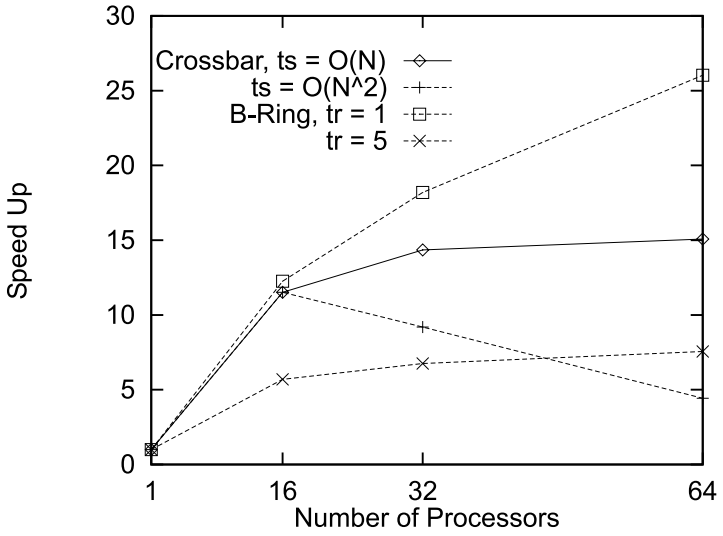


Fig. 3. Speed up/FFT.

ring a write miss generates an invalidation message that traverses the entire ring and hence increases the ring utilization. With fast ring speed ($t_r = 1$), the bidirectional ring outperforms the crossbar by 6%, 27%, and 73% for $N = 16$, 32 and 64 and $t_s = 4$, 8, and 16, respectively.

Miss rate of Simple increases for larger N while local miss rate decreases. These cause more network traffic and hence are disadvantage for both network. On the other hand, decreasing write miss rate and increasing nearest neighbor communication pattern of Simple are advantageous for the bidirectional ring. With $t_r = 1$, the bidirectional ring outperforms the crossbar by 5%, 19%, and 70% for $N = 16$, 32 and 64 and $t_s = 4$, 8, and 16, respectively (Fig. 4). When $N = 32 \rightarrow 64$, the bidirectional ring achieves speed up of 40% while the performance of the crossbar decreases slightly even with an assumption of $O(N)$ communication latency growth rate (Fig. 5).

Weather has relatively low miss rate, and hence the performance difference between two networks is relatively small, especially for $N = 16$ and 32 (Fig. 6). However, when N is increased from 32 to 64, the effect of interconnection network becomes stronger due to increasing miss rate by 37% and decreasing local miss rate by 40%. The bidirectional ring can still achieve speed up of 46% with $t_r = 1$, while the performance of the crossbar stays almost the same (Fig. 7). The sources of this performance difference between two networks are considered to be decreasing write miss rate by 20% and increasing degree of nearest neighbor communication for $N = 32 \rightarrow 64$. With $t_r = 1$, the bidirectional ring outperforms the crossbar by 2%, 9% and 58% for $N = 16$, 32 and 64 and $t_s = 4$, 8, and 16, respectively.

The miss rate and local miss rate of Speech are constant over increasing N. Therefore, the growth of communication latency of interconnection networks is the dominant factor for the performance. In addition, since Speech has the all-to-all communication pattern, relative performance of Speech is mainly affected by the speed of both networks (Fig. 8). With $t_r = 1$, the bidirectional ring outperforms the crossbar by 5%, 16%, and 45% for $N = 16$, 32 and 64 and $t_s = 4$, 8, and 16, respectively. For $N = 32 \rightarrow 64$, the bidirectional ring still provides some level of

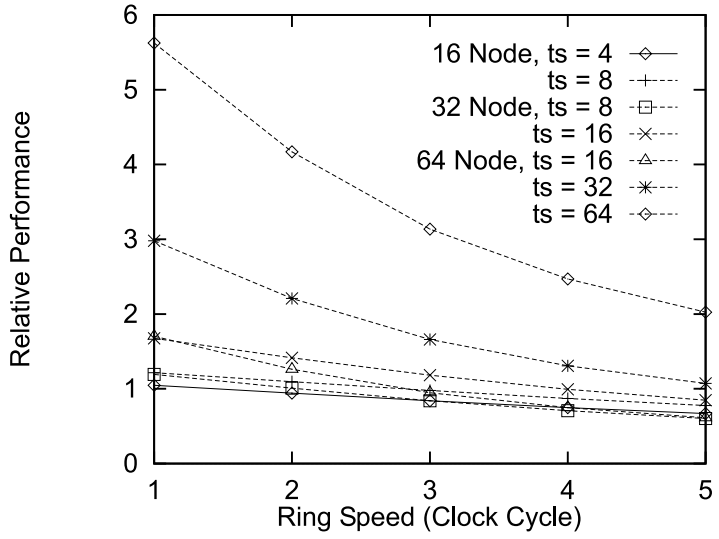


Fig. 4. Relative performance/Simple.

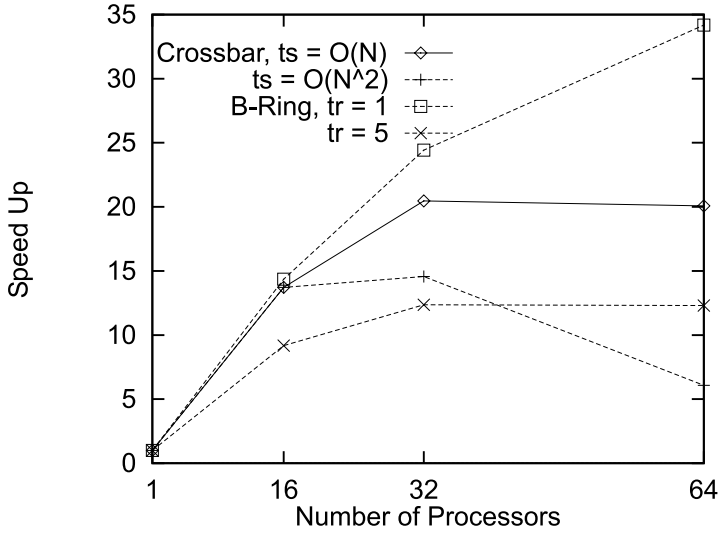


Fig. 5. Speed up/Simple.

speed up (18% to 66%), while the performance of the crossbar 7 can be either increased by 32% or decreased by 42% depending on the speed of the switch (Fig. 9).

The average (geometric mean) of relative performance and speed up of both networks are shown in Figs. 10 and 11. On the average of four applications, the bidirectional ring provides better performance by 4%, 21%, and 61% for $N = 16$, 32 and 64 and $t_s = 4$, 8, and 16, respectively.

The slowest ring speed for the bidirectional ring (Figs. 12–14) to outperform the crossbar are shown in Table 3. For $N = 16$, we need a very fast ring that can operate at the same speed as the

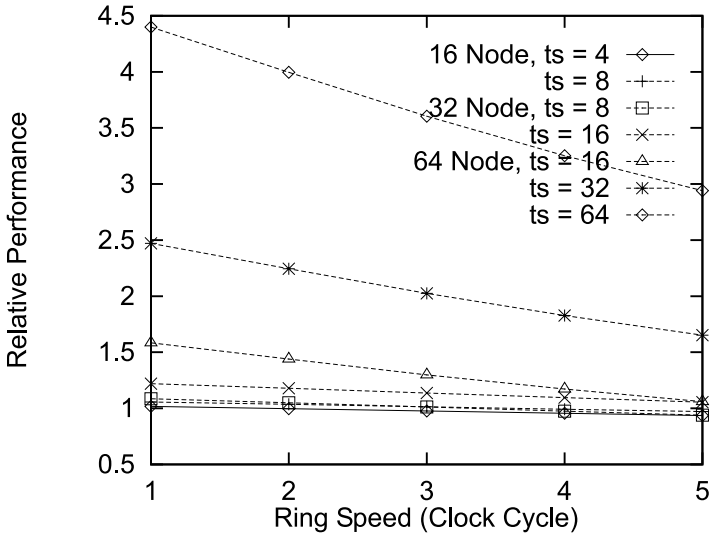


Fig. 6. Relative performance/Weather.

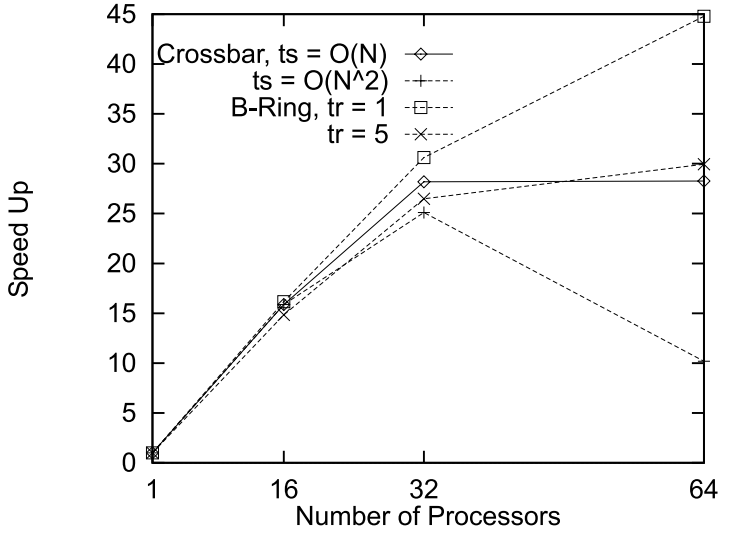


Fig. 7. Speed up/Weather.

processor clock ($t_r = 1$) to outperform a very fast implementation of a crossbar switch network ($t_s = 4$). If the speed of a crossbar switch is moderate, a ring that operates at half the speed of the processor clock is sufficient. For $N = 32$, a bidirectional ring with half and one fourth the speed of the processor clock suffice to outperform the crossbar switch network with very fast and moderate ($t_s = 8$ and 16) speed respectively. For $N = 64$, it is much easier for a bidirectional ring to out-

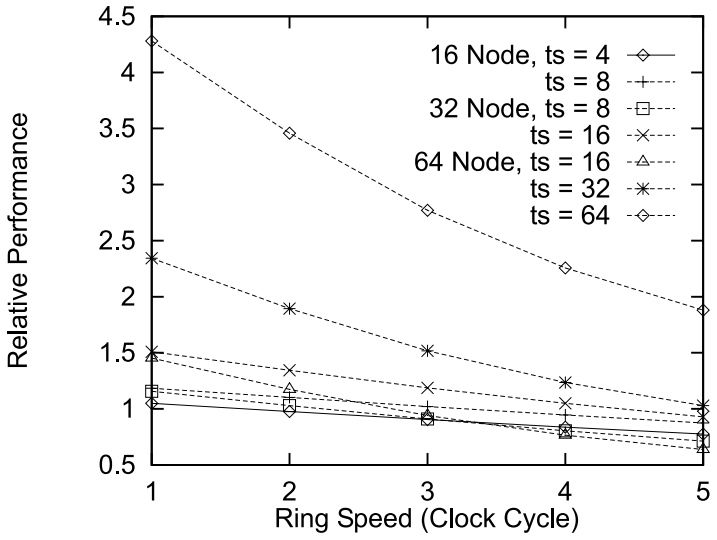


Fig. 8. Relative performance/Speech.

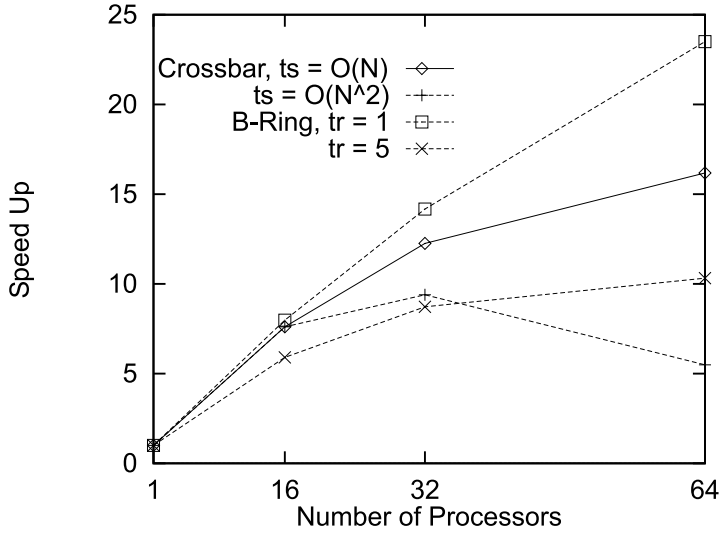


Fig. 9. Speed up/Speech.

perform a crossbar switch network due to the latter physically slow operating speed. However, although it outperforms the crossbar, without sufficiently fast ring speed, we cannot expect speed up on the bidirectional ring. For example, on the average of four applications, we have speed up of 48% on the bidirectional ring with $t_r = 1$ when N is increased from 32 to 64. On the other hand, if $t_r = 5$, which is sufficient to outperform a moderately fast crossbar, the speed up is only 11% for $N = 32 \rightarrow 64$.

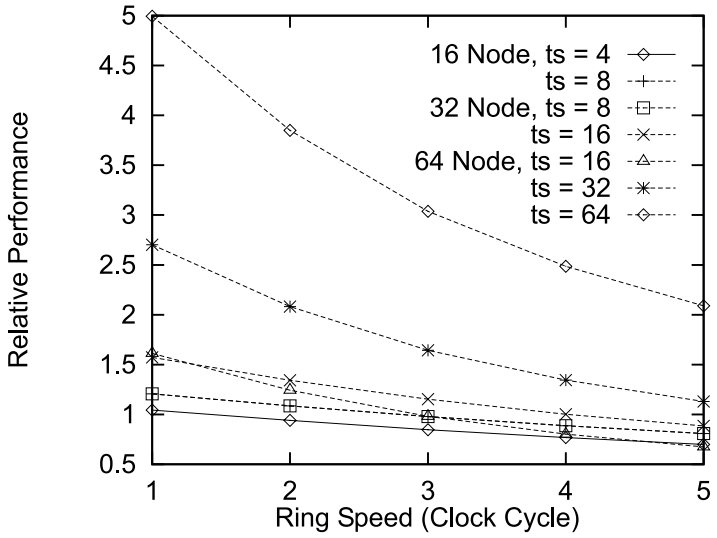


Fig. 10. Relative performance/Average.

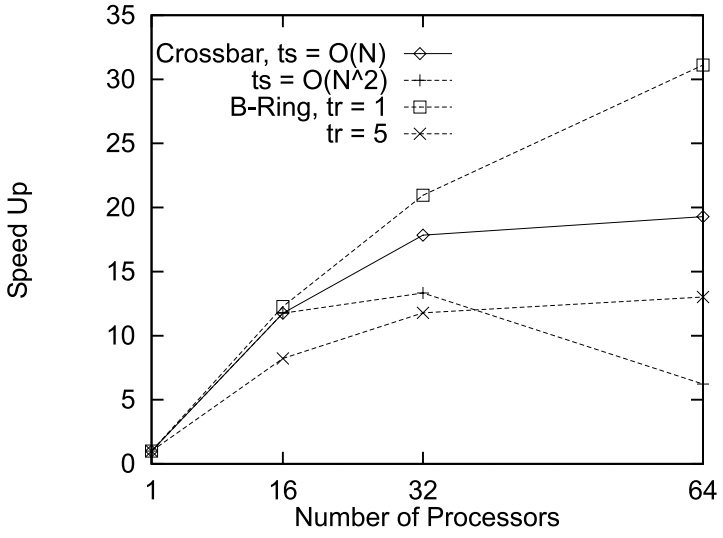


Fig. 11. Speed up/Average.

5. Conclusions

In this paper, we have evaluated the performance of the bidirectional ring by comparing it to the performance of the crossbar network. We used a hybrid evaluation method which is a combination of an analytical model and workload parameters extracted from the memory traces of four parallel applications. Our study indicates that for a 16 nodes configuration, which is a

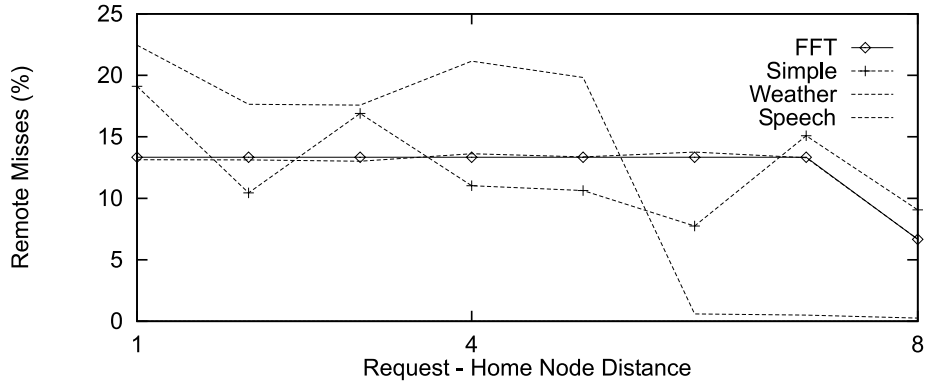


Fig. 12. Home node distribution (bidirectional ring, 16 nodes).

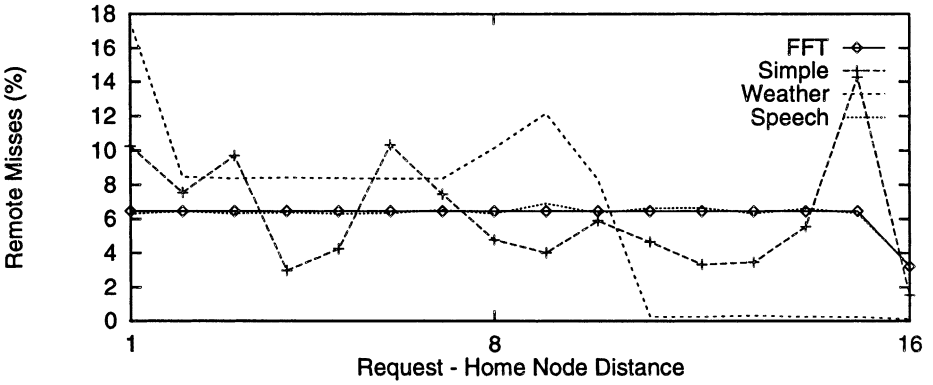


Fig. 13. Home node distribution (bidirectional ring, 32 nodes).

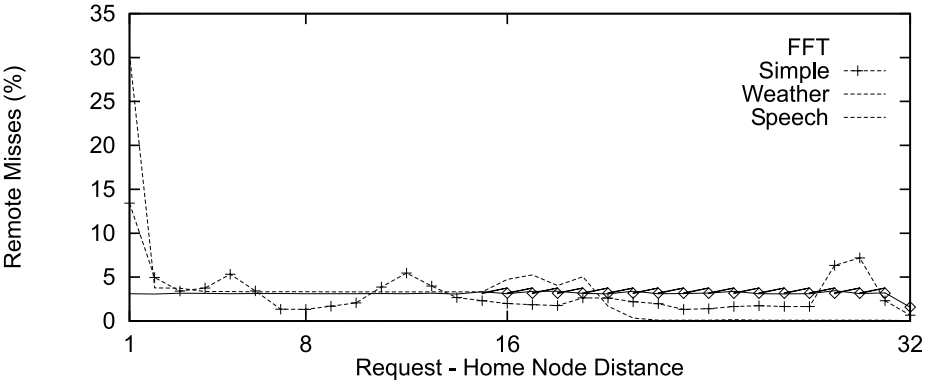


Fig. 14. Home node distribution (bidirectional ring, 64 nodes).

Table 3
Ring speed to outperform crossbar

N	t_s	Application				Average
		FFT	Simple	Weather	Speech	
16	4	1	1	1	1	1
	8	2	2	3	3	2
32	8	1	2	3	2	2
	16	3	3	>5	4	4
64	16	2	2	>5	2	2
	32	4	>5	>5	>5	>5
	64	>5	>5	>5	>5	>5

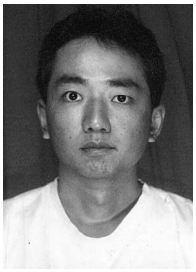
typical size of crossbar switches used in some of actual multiprocessors, both architecture achieve similar performance for the link speed we have assumed. For a 32 nodes configuration, the bidirectional ring outperforms the crossbar by 21% on the average of four application programs, with an assumption that the growth rate of the communication latency of the crossbar is suppressed to $O(N)$. For a 64 nodes configuration, we can still expect speed up on the bidirectional ring (48% on the average) while that of the crossbar increases only by 8%.

Further investigations could include analysis of dynamic behavior of both networks (such as hot spot contention) using execution-driven simulations and the hierarchical network models.

References

[1] Oi H, Ranganathan N. Performance analysis of the bidirectional ring-based multiprocessor. Proceedings of the ISCA Tenth International Conference on Parallel and Distributed Computing Systems, 1997. p. 397–400.
[2] Oi H, Ranganathan N. Effect of message length and processor speed on the performance of the bidirectional ring-based multiprocessor. Proceedings on the International Conference on Computer Design, October 1997. p. 267–72.
[3] Lang T, Valero M, Alegre I. Bandwidth of crossbar and multiple-bus connection for multiprocessors. Trans Comp IEEE 1982;c31(12):1227–34.
[4] Ravindran G, Stumm M. A performance comparison of hierarchical ring- and mesh-connected multiprocessor networks. Proceedings of International Symposium on High Performance Computer Architecture, February 1997. p. 58–69.
[5] Barroso LA, Dubois M. Performance evaluation of the slotted ring multiprocessor. Trans Comp IEEE 1995;44(7):878–90.
[6] Vranesic Z, et al. The NUMAchine Multiprocessor. Technical report, Department of electrical and computer engineering, Department of computer science, University of Toronto, June 1995.
[7] Kendall square research corporation. KSR1 Technical Summary, 1992.
[8] Exemplar System Architecture. http://www.hp.com/wsg/products/servers/exemplar/sx-class/exemplar2.html, Hewlett Packard.
[9] Boku T, et al. Architecture of massively parallel processor CP-PACS. Proceedings of the 1997 Second Aizu International Symposium on Parallel Algorithms/Architecture Synthesis, Fukushima, Japan, 1997. p. 31–40.
[10] Sterling T, et al. An empirical evaluation of the convex SPP-1000 hierarchical shared memory system. Int J Parallel Prog 1996;24(4):377–96.

[11] Gustavason DB. Scalable coherent interface and related standards projects. IEEE MICRO 1992;12(1):10–22.
[12] Zhang X, Yan Y. Comparative modeling and evaluation of CC-NUMA and COMA on hierarchical ring architecture. Trans Parallel Distributed Sys IEEE 1995;6(12):1316–31.
[13] Farkas K, Vranesi Z, Stumn M. Scalable cache consistency for hierarchically structured multiprocessors. J Supercomput 1995;8:345–69.
[14] Davis H, Goldschmidt SR, Hennessy R. Multiprocessor simulation and tracing using tango. Proceedings of the 1991 International Conference on Parallel Processing, vol. II, 1991. p. 99–107.
[15] Bhuyan L, Ghosal D, Yang Q. Approximate analysis of single and multiple ring networks. Trans Comp IEEE 1989;38(7):1027–40.
[16] Chaiken D, et al. Directory-based cache coherence in large-scale multiprocessors. IEEE Comp 1990;23(6):49–58.
[17] ftp://tracebase.nmsu.edu/pub/, Parallel Architecture Research Laboratory, New Mexico State University.



Hitoshi Oi is an assistant professor in the Department of Computer Science and Engineering at the Florida Atlantic University. Before joining FAU, he was working with HAL Computer Systems in Campbell, California, as a Computer Architect. He received a Ph.D. degree in Computer Science and Engineering from the University of South Florida. He also reveived a Bachelor's degree in Electronics and Communication Engineering and a Master's degree in Electrical Engineering, both from Meiji University. His research interests include computer architecture and performance evaluation, LSI design and verification, and non-standard logic.

N. Ranganathan received the Ph.D. degree in Computer Science from the University of Central Florida, Orlando in 1988. He is currently a professor in the Department of Computer Science and Engineering and the Center for Microelectronics Research at the University of South Florida, Tampa. His research interests include VLSI design, design automation, hardware algorithms, computer architecture and parallel processing.