Effect of Message Length and Processor Speed on the Performance of the Bidirectional Ring-Based Multiprocessor

Hitoshi Oi and N. Ranganathan
Department of Computer Science and Engineering,
University of South Florida
Tampa, FL 33620
Email: {oi, ranganat}@csee.usf.edu

Abstract

This paper presents a comparative study of the performance of the bidirectional ring and the unidirectional ring multiprocessor, with emphasis on the effect of system parameters, specifically, the message length and the relative processor speed. The choice of these parameters may not be optimum due to the performance cost tradeoffs in practice. Our study shows that the use of bidirectional ring is more effective in such suboptimum system configurations and can improve the processor utilization by up to 35%.
Keywords *Interconnection networks, distributed shared memory architecture, performance evaluation, slotted ring.*

1 Introduction

In a distributed shared memory (DSM) multiprocessor, a globally shared address space is provided while the memory units are physically distributed among processing elements (PE). The PE's are connected by an interconnection network and the consistency of multiple copies of the data is maintained by sending coherence control messages over the network. Interconnection networks that have been used in commercial/research multiprocessor systems include the 2-D Mesh (Stanford FLASH [1]) and the fat-tree (TMC CM-5 [2]). The ring interconnection network has 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).

NUMAchine [3] is a multiprocessor developed at the University of Toronto. It has three hierarchy levels: a small number of PE's connected by a bus form a cluster and the clusters are connected by two levels of rings. Holiday and Stumm investigated the performance of the hierarchical ring-based multiprocessor using parametric simulations [4]. Barroso and Dubois evaluated the performance of the slotted ring multiprocessor using a hybrid approach (combining trace-driven simulations and analytical models) [5] . KSR-1 of Kendall Square Research was a cache-only memory architecture (COMA) multiprocessor with hierarchy rings [6]. Scalable Coherent Interface (SCI) defined by IEEE P1596 standard also provides ring interconnection networks for DSM multiprocessors [7].

Most research in the past including the above are based on unidirectional rings. In a unidirectional ring, the messages have to traverse all the way through the ring even if the destination is within the local neighborhood. The length of traversal can be significantly reduced by using bidirectional rings. In this paper, we analyze the performance of the bidirectional slotted ring by comparing it to the performance of the unidirectional ring architecture, with emphasis on the effect of the following system parameters: the data message length and the relative speed between the processor and other components. By developing a detailed analytical model, the performance improvement is quantitatively studied.

The rest of the paper is organized as follows. The architectures of the ring-based multiprocessors that are assumed in our study are described in the next section. The analytical model for performance evaluation is derived in Section 3. In Section 4, the performance of the bidirectional ring architectures and the influence of the system parameters are studied in terms of access latency and processor utilization. Some conclusions are provided in Section 5.

2 Architecture Model

The architecture of the ring-based multiprocessor and its node are shown in Figure 1. We assume the use of slotted rings to calculate the latency of the network. The number of data lines (say m) that connect the neighboring PE's are assumed to be the same in the unidirectional and bidirectional rings. Thus, a packet on the unidirectional ring is equivalent to two packets on the bidirectional ring. Data messages and header messages on the bidirectional ring are more than one packet long, and hence they are divided into multiple packets and transmitted separately. Each PE consists of a processor, a cache, a memory unit and a network interface. It is assumed that the architecture is based on the CC-NUMA model with directory based cache coherency protocol. The local memory 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. If a cache miss corresponds to a non-local memory address, it will be first sent to the home node (the PE that is assigned the portion of global address of the accessed data), and the home node responds with a data message unless the requested data is dirty. If the requested data is dirty, the read request is forwarded to the owner PE (the PE that owns a modified copy of the requested data in the cache), and the owner PE responds with the data. An invalidation message is broadcast by passing it through the PE's on the ring instead of sending multiple invalidation messages. In the bidirectional ring, the network interface selects the link corresponding to the shorter path to the destination of the message.

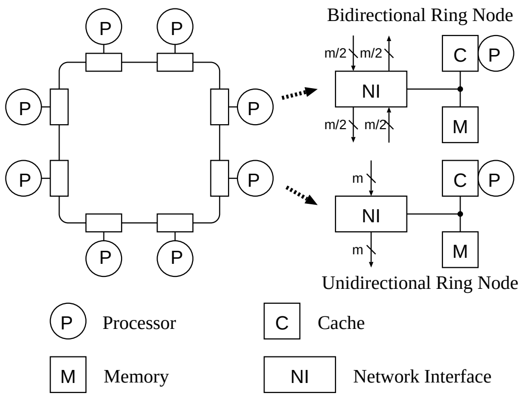

Figure 1. Ring-Based Multiprocessor Model

The parameters that define the configuration of the system are listed in Table 1. The table also shows the defaults values (underlined) and the variations of these of parameters that will be used in Section 4 to see their effect. The number of processors, N, is assumed to represent a medium configuration as we assume single level ring networks. In this study we assume $N = 32$ as in the case of KSR-1. The header messages (read request and invalidation) are of length one packet in the unidirectional ring and two packets in the bidirectional ring, while the data message consists of dp times as many packets as a header message. The latency corresponding to the transfer of a packet to the adjacent node is represented by t_r. Each memory access t_m corresponds to 30 (45, 60) processor clock cycles. The time for cache protocol handling, t_c, is assumed to be 2 (3, 4) processor clock cycles. A cache hit corresponds to one processor cycle. The above assumptions are taken from previous studies including [4]. The processor is blocked on a cache miss.

Symbol	Definition	Assumption
N	No. of PE's	32
t_r	Ring speed	<u>2</u>, 3, 4 cycles
t_m	Memory access time	<u>30</u>, 45, 60 cycles
t_c	Cache protocol	<u>2</u>, 3, 4 cycles
dp	Message length ratio	1, 2, 4, <u>8</u>

Table 1. System Parameters (Default values are underlined)

3 Analytical Model

In this section, we derive analytical models of both the unidirectional and bidirectional slotted ring architectures for performance evaluations. The workload parameters that represent the behavior of programs are listed in Table 2. The locality of data access can be classified as (i) *local*: accesses to local memory within the PE (ii) *group*: accesses to addresses in PE's within the distance G (within G hops) and (iii) *global*: all other accesses. It is assumed that the accesses within the same class are uniformly distributed. The cache misses occurring in a PE are represented by Poisson distribution. The cache miss rate, MR, is the probability of the cache miss per reference (the processor blocking time is not included). The cache misses are divided as P_L to be local, $(1 - P_L)P_G$ to be group and $(1-P_L)(1-P_G)$ to be global. The above model is a simplification of the cluster model used in [4] except that the number of hops is limited in this model whereas the model in [4] only limits the number of PE's.

Symbol	Parameters
G	Group size
MR	Cache miss rate
P_W	Prob. of write access
P_L	Prob. of local mem access
P_G	Prob. of group mem access

Table 2. Workload Parameters

3.1 Access Classification

The classification of non-local accesses is given in Figure 2. The classification tree shown is based on the access modes and access classes. The probabilities of accesses for different classes and modes are indicated as labels on the edges. In the case of a unidirectional ring, the branches with group and global classes are not applicable. The effective communication distance ECD is the average message traversal length of non-local cache miss and is weighted by the probabilities on the corresponding leaf nodes. The probabilities for write and read accesses are represented as P_W and $1 - P_W$ respectively. A write access involves either fetching the block or broadcasting an invalidation message.

We approximate the state of a memory block, either clean or dirty, as follows. With a sufficiently large cache size and long execution time, the effect of cold and conflict are negligible. Thus, the effect of invalidation misses is dominant. The first read access to a memory location which was written by some other PE causes a read miss to a dirty block, but it also writes the dirty block back to the memory. Hence, the state of the memory block becomes clean after the first read miss. Therefore, we assume that probabilities that a block is clean and dirty state are $(1 - 2P_W)/(1 - P_W)$ and $P_W/(1 - P_W)$ respectively. In a bidirectional ring, the destination PE of a read request can be either within the group or global with probabilities P_G and $1 - P_G$ respectively. In the case of a read to a dirty block, the owner can be either within the group (P_G) or global $(1 - P_G)$.

A break down of the latency of the access modes is given in Figure 3. The *transmission delay* T is the time for a PE to transmit a message (read request, data, or invalidation). The *propagation delay* P is the time for a message to reach the destination PE and is determined by the number of hops. By definition, the weighted sum of all the P's appearing in Figure 2 is the ECD described above. The *memory access latency* M is the memory access time t_m defined in the previous section plus the average waiting time in the memory access queue. The *protocol handling latency* C is the time to look up the directory, access a cache block from the network interface and prepare to transmit, or process the received data message and restart the processor that was stalled on a miss. This is equal to t_c defined in the previous section.

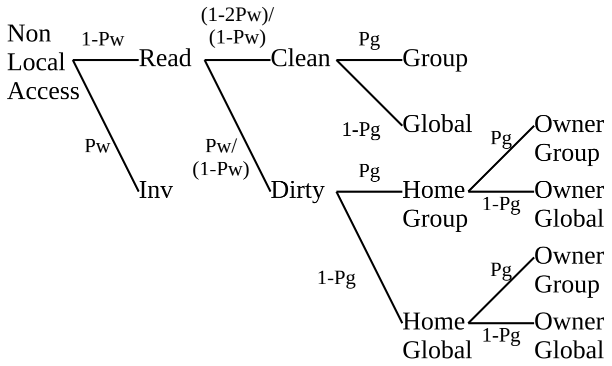

Figure 2. Classification of Non-local Access

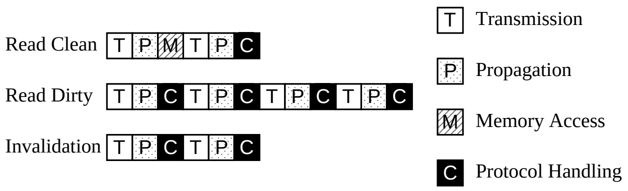

Figure 3. Access Mode Latencies

3.2 Miss Latency and Processor Utilization

The derivations for miss latency L and miss rate MR are as follows: First, we set the miss frequency MF that is the cache miss per processor clock cycle. It should be noted that the miss frequency is different from the miss rate MR which is the number of memory accesses per reference - the period during which the processor is blocked *is* included in MF. To calculate the transmission delay T on the slotted ring, we use approximated $M/G/1$ model by Bhuyan et al in [8] with slight modification to suit to our problem. The transmission delay T consists of W and x, where W is the mean waiting time for a message in the transmission queue and x is the number of packets in the message. We explain the simplest case: the transmission delay of a header message, that consists of a single packet, on the unidirectional ring here. We will explain the case of longer data message in Section 4.1. Since we assume that a packet is transmitted in a single ring clock cycle, $x = t_r$. W consists of α (the time to wait for the beginning of the slot, $= t_r/2$), d (the time to find an empty slot on the ring) and the time to transmit the messages ahead in the queue ($Q(d+x)$, where Q is the number of messages ahead in the transmission queue, and is equal to λW).

Now, $d = U/(1-U)$ where U is the ring utilization. We assume that successive slots behave independently as the error produced by this approximation is known to be relatively small [8]. U is given by $\lambda ECDx$ where λ is the average message transmission rate. Since only the non-local misses transmit messages on the ring, $\lambda = (1-P_L)MF$. The time to access the main memory M is t_m plus the time to complete the remaining accesses in the memory queue. Similar to the calculation of d, we get $M = (1 + Mu/(1-Mu))t_m$, where memory utilization $Mu = MF(1-P_W)t_m$ since in our model the main memory is accessed only by the read misses.

Thus, the access latency L can be derived as:

$$\begin{aligned} L &= P_L M + (1-P_L)\{2(1+P_W)T + ECD \\ &\quad +(1-2P_W)M + (1+4P_W)C\} \end{aligned}$$

Next, we obtain the miss rate MR and the expression for processor utilization u from miss latency L. Since the processor blocks on a miss, we have

$$1/MF = 1/MR + L.$$

The processor performs either computation or data access from the cache for the period of $1/MR$ on the average, and then it stalls for the period L due to a cache miss. Thus, from miss frequency MF, we get processor utilization $u = 1 - MFL$ and miss rate $MR = MF/(1-MFL)$.

4 Performance Comparison

In this section, we compare the performance of the unidirectional and the bidirectional ring and show how the advantage of the bidirectional ring is affected by varying system parameters. We use the write probability $P_W = 0.3$ as in the related study including [4] and the group size $G = 1$ by assuming nearest neighbor communication in the performance evaluations below.

4.1 Effect of Data Message Length

Past studies including [9, 10] pointed out that unlike uniprocessor systems, the larger cache block size is not beneficial for multiprocessor systems due to the false sharing miss, the increased data traffic, and the smaller degree of the spatial locality. Nevertheless, large cache block sizes are used for multiprocessor systems in practice. One of the reasons is to reduce the hardware overhead to store the tag and the state information of cache blocks. For instance, a cache block size of 128 bytes is used for the Stanford FLASH [1] and the KSR-1 [6][1].

Miss Mode	Proba -bility	Header (1 packet)	Data (dp packets)
Clean	$1-2P_W$	1	1
Dirty	P_W	2	2
Inval	P_W	2	none

Table 3. Access Mode & Packet Transmission

In this section, the effect of long data messages on the performance is studied. As the size of the cache block increases we also need to increase the length of the data message. As a sample case, we use $P_L = P_G = 0.5$. One way to accommodate a longer data message used in [5] is to format the entire ring into frames each of which consists of slots for a combination of header messages and a data message. However this approach could lead to a lower utilization of the ring because each slot can only accommodate a designated message type. Another approach is to divide a data message into multiple packets and transmit them independently. The advantage of this approach is that a packet can be transmitted whenever an empty slot is available, while the disadvantage is that each packet must include information to reassemble a message from packets. In this paper, the latter approach is assumed.

Different type of miss involves different number of header and data message transmissions. Probability and numbers of header and data messages for each type of access mode are summarized in Table 3. From this table, the average number of packets per miss (read miss or invalidation) is $1 + 2P_W + dp$ and the average number of message transmission per miss is $2+2P_W$. Thus, the average number of packets per message transmission is $(1+2P_W+dp)/(2+2P_W)$ (doubled on the bidirectional ring).

A larger block size also increases the number of words accessed from the memory unit per read miss, and in turn increases the memory utilization. We assume that to transmit a data message of the packet length dp, the memory unit is busy for the period of $t_m + (dp-1)t_r$.

The transmission delays for the different data message sizes are plotted in Figure 4. The ring utilization

[1]The size of the subpage in KSR-1's ALLCACHE (COMA) memory system.

of the bidirectional ring is lower than that of the unidirectional ring since the message always traverses the shorter path on the ring. Consequently on the bidirectional ring the transmission delay per packet is lower. However, since we assume that the total number of data lines for the link for a PE are the same on both rings (i. e., one packet on the unidirectional ring is equivalent to two packets on the bidirectional ring), the transmission delay per message is doubled on the bidirectional ring. As a result the transmission delay per message is higher on the bidirectional ring for $dp = 1, 2$ (not shown) and for $dp = 4$ with low miss rate. However for $dp = 8$, the unidirectional ring is almost saturated (ring utilization is about 86% for $MR = 0.05$, while that of the bidirectional ring is 52%), and exhibits much higher transmission delay than the bidirectional ring. Also the transmission delay of the unidirectional ring grows faster against the increase of the miss ratio. As a result, the ratio of processor utilization between bidirectional and unidirectional rings (PUR) jumps up at $dp = 8$ (Figure 5).

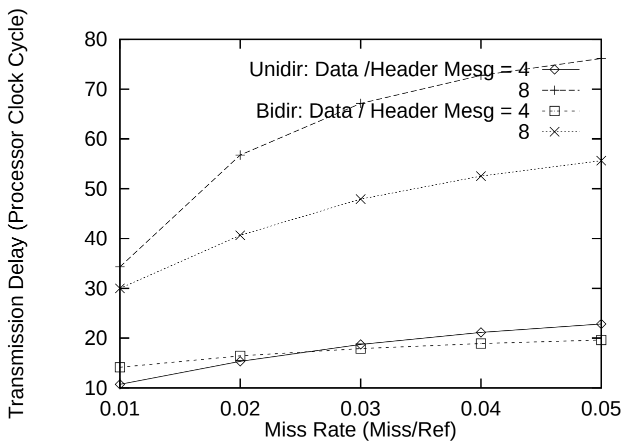

Figure 4. Transmission Delay Per Message

4.2 Effect of Relative Speed

The relative speed between the processor and other components (ring, protocol handling at the network interface and memory access) has been fixed so far. As the implementation technology of the microprocessor progresses, the speed gap between the processor and other components is expected to increase. In this section we will look at how the relative speed affects the performance of the ring-based multiprocessor and how the advantages of the bidirectional ring changes.

We use the following cases represented by triples of components' latencies relative to the processor's speed

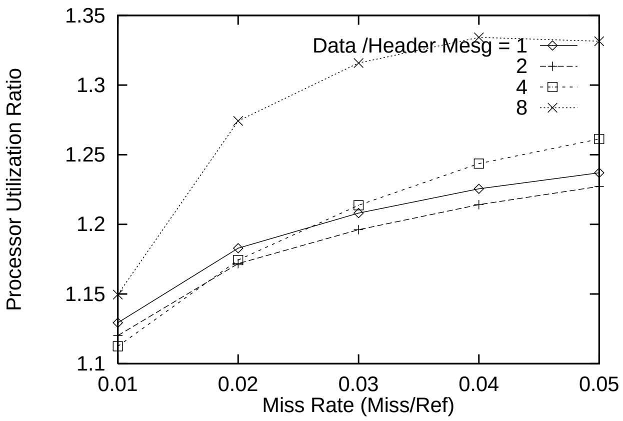

Figure 5. PUR for Longer Data Message

(t_r : t_c : t_m). Case i (2 : 2 : 30) is the base case that has been used in our study so far. In case ii (3 : 3 : 45) and case iii (4 : 4 : 60), the latencies of all the components are increased by 50% and 100% while the ratios between them are unchanged. These two cases correspond to the situation where faster processors become available and they are used with the same memory system. When we construct a multiprocessor system with off-the-shelf components, the most expensive part is expected to be the network interface, because it has much less market demand than the processor and memory components. To compromise the cost and the performance, we may use FPGAs or other programmable devices. Case iv (4 : 4 : 30), in which t_r and t_c are doubled while t_m is the same as case i, represents such an implementation of the multiprocessor system.

Config	Case i	Case ii	Case iii	Case iv
Unidir	0.477 (1.000)	0.345 (0.724)	0.270 (0.565)	0.273 (0.573)
Bidir	0.549 (1.000)	0.428 (0.780)	0.346 (0.630)	0.361 (0.657)

Table 4. Effect of Relative Speed on Processor Utilization ($MR = 0.01$)

The processor utilization for each case is shown in Table 4 ($P_L = P_G = 0.5$ and $dp = 8$ are used. Due to space limitation, only the figures for $MR = 0.01$, in which the variation among cases is the largest, are shown). To see how slower memory systems degrade the performance, the processor utilization normalized to that of case i with the same ring is added with parenthesis below the processor utilization. It is unavoidable for both unidirectional and bidirectional ring to suffer from the slower memory systems. However, the use of the bidirectional ring can mitigate this effect. For example, in case iv, in which the ring and the network interface operate at half the speed, the processor utilization of the unidirectional ring is decreased to 57% of case i while that of the bidirectional ring is 66% of case i. From the results shown in Figures 4 and the PUR in Figure 6, the bidirectional ring is more effective (1) when the miss rate is low and the relative speed gap is large, or (2) the miss rate is relatively high (Figure 6).

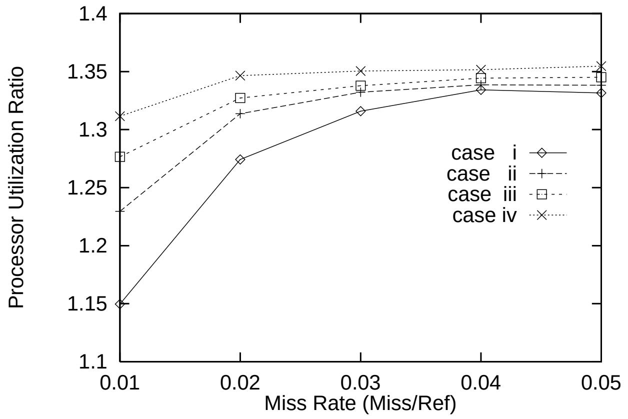

Figure 6. Effect of Relative Speed on PUR

5 Conclusions

In this paper, we have presented the comparison of the unidirectional ring and the bidirectional ring under varying system parameters using an analytical model. In practice, the data message is likely to be much longer than the header message. This would increase the transmission delay due to higher ring utilization. Similarly, the relative speed between the processor and other components may not be completely balanced due to various reasons such as technology, cost-performance tradeoffs, etc. However, the degradation of the system performance can be alleviated by the use of the bidirectional ring. For example, when the data message is eight times longer than the header message, that is the case in KSR-1, the processor utilization of the bidirectional is better than the unidirectional ring by 15% to 33%. Further investigations could include the extraction of workload parameters from applications and the analysis of dynamic system behavior using execution-driven simulations.

Acknowledgment

This research is supported in part by a National Science Foundation Grant No. CDA-9522265.

References

[1] Jeffrey Kuskin et al,, *The Stanford FLASH Multiprocessor*, In Proceedings of the 21st International Symposium on Computer Architecture, 302–313, April 1994.

[2] C. Leiserson et al., *Network architecture of the Connection Machine CM-5*, In Proceedings of 4th Annual ACM Symposium on Parallel Algorithms and Architectures, 272–285, June–July 1992.

[3] Z. Vranesic et al, *The NUMAchine Multiprocessor*, Technical Report, Department of Electrical and Computer Engineering, Department of Computer Science, University of Toronto, June 1995.

[4] M. Holiday and M. Stumm, *Performance of Hierarchical Ring-Based Shared Memory Multiprocessors*, Transactions on Computers, IEEE, Vol. 43, No. 1, 52–67, January 1994.

[5] L. A. Barroso and M. Dubois, *Performance Evaluation of the Slotted Ring Multiprocessor*, Transactions on Computers, IEEE, Vol. 44, No. 7, 878–890, July 1995.

[6] Kendall Square Research Corporation, *Technical Summary*, 1992.

[7] D. B. Gustavason, *Scalable Coherent Interface and Related Standards Projects*, IEEE MICRO, Vol. 12, No. 1, 10–22, February 1992.

[8] L. Bhuyan, D. Ghosal and Q. Yang, *Approximate Analysis of Single and Multiple Ring Networks*, Transactions on Computers, IEEE, Vol. 38, No. 7, 1027–1040, July 1989.

[9] A. Gupta and W-D. Weber, *Cache Invalidation Patterns in Shared-Memory Multiprocessor*, Transactions on Computers, IEEE, Vol. 41, No. 7, 794–810, July 1992.

[10] J. Torrellas, M. S. Lam and J. L. Hennesy, *False Sharing and Spatial Locality in Multiprocessor Caches*, Transactions on Computers, IEEE, Vol. 43, No. 6, 651–663, June 1994.