Proc. Intl. Conf. Parallel Processing, pp.I/51-55, 1995.

Hierarchical Ring Topologies and the Effect of their Bisection Bandwidth Constraints

G. Ravindran and M. Stumm
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario, Canada M5S 1A4
Email: ravin@eecg.toronto.edu

Abstract -- *Ring-based hierarchical networks are interesting alternatives to popular direct networks such as 2D meshes or tori. They allow for simple router designs, wider communications paths, and faster networks than their direct network counterparts. However, they have a constant bisection bandwidth, regardless of system size. In this paper, we present the results of a simulation study to determine how large hierarchical ring networks can become before their performance deteriorates due to their bisection bandwidth constraint. We show that a system with a maximum of 128 processors can sustain most memory access behaviors, but that larger systems can be sustained, only if their bisection bandwidth is increased.*

1.0 Introduction

Shared memory multiprocessors based on hierarchical ring networks such as those for Hector [9], KSR [1], and the NUMAchine [10] are interesting alternatives to those based on popular direct networks such as 2D meshes or tori. Their simple node to ring interface allows them to be clocked at a much faster rate and the smaller number of connections at each node allow for wider data paths. An important parameter of an interconnection network is its bisection bandwidth, which is defined as the bandwidth provided by the minimum number of wires cut when the network is divided into two equal halves [2]. The bisection bandwidth of a hierarchical ring network is less when compared to a k-ary 2-cube network of equal size, and more importantly the bisection bandwidth is constant and does not scale with the size of the network. This raises the question of how large ring-based multiprocessors can become and what the best topologies are, given the constraint on the bisection bandwidth.

We use a bottom-up approach to address this question. First, we start with a single, *local* ring and determine how many nodes it can accommodate before the performance of the ring deteriorates. We then consider two level hierarchies and determine how many such local rings can be connected to a second level ring while still maintaining reasonable performance. We then proceed to determine how many such second level rings can be sustained by a third level and so on. We show that the bisection bandwidth constraint severely affects the performance, allowing at most three levels of hierarchy and approximately 128 processors, unless there is a great deal of memory access locality. We also explore how sensitive the performance of the network is to its bisection bandwidth. For example, an interesting observation we make is that increasing bisection bandwidth can, at times, also hurt performance by increasing congestion at the local-ring level.

We use a detailed flit-level simulator to study these issues. We simulate hierarchical blocking networks with register insertion rings, where the processor-to-ring and ring-to-ring interfaces are similar to that of SCI [3] node interfaces[1]. The simulator is driven by a synthetic micro-benchmark that generates memory reference sequences with different access and sharing patterns. Thus we are able to emulate a wide spectrum of memory access behaviors, from high to low cache miss rates and from high degrees of memory locality to almost no locality.

There have been only few studies on performance of scalable hierarchical ring networks so far. Holliday and Stumm [5] studied the performance of large scale hierarchical slotted ring architectures. Throughout their study they assumed very large degrees of locality in their workloads which makes their results applicable only for well-behaved applications. Hamacher and Jiang [4] used analytical models and compared the performance of hierarchical ring interconnects with 2D meshes and concluded that a 3-level hierarchical ring performs somewhat better than a 2D mesh.

2.0 Simulated System

2.1 System Description

Figure 1 shows a two-level hierarchical ring interconnect. It consists of processing modules (PM) connected by a hierarchy of unidirectional rings. A processing module is connected to the lowest level *local* ring and contains a processor, a local cache, and a part of the main memory. A highest-level *global* ring connects several of these local rings. This is similar to the Hector architecture [9]. The system provides a flat, global address space and each PM is assigned a unique contiguous portion of that address space, determined by its location. All processors can transparently access all memory locations in

1. Our channel width and packet sizes are different from SCI standard. Also, we do not model SCI bandwidth allocation and queue allocation protocols.

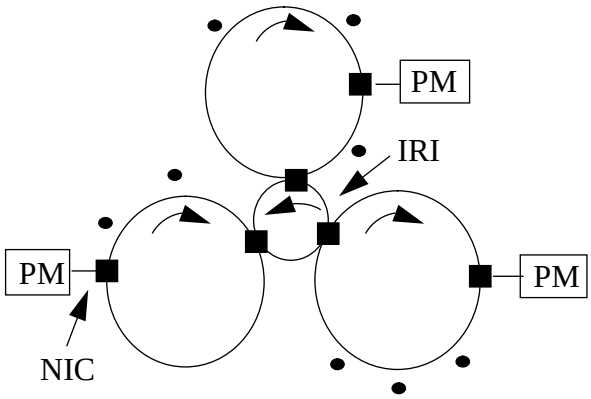

Figure 1: Two Levels of Ring Hierarchy

the system. Local memory accesses do not involve the network. Remote memory accesses require that a request packet be sent from the requesting processor to the target memory, followed by a response packet from the target memory to the requesting processor.

The packets are of variable size and are transferred in a bit-parallel format along a unique path through the ring hierarchy. If a packet is larger than the width of the ring, then it is sent as a contiguous sequence of flits, with the header flit containing routing and sequencing information. We assume *Wormhole* Routing, where a packet may become spread across contiguous links of the network as flits are forwarded, but the packet cannot be interleaved with the flits of other packets [2].

There are five main types of packets, namely read and write requests, read and write responses and negative acknowledgment packets. If a requesting module receives a negative acknowledgment in response to one of its requests, then it resends the request after a short delay. In our simulations, we assumed a cache line size of 16 bytes and that a processor can have at most 3 outstanding (prefetching) requests. The channel is assumed to be 128 bits wide and based on the NUMAchine multiprocessor [10]. Read responses and write requests require 256 bit packets (two flits), whereas read requests and write responses require 128 bit packets (single flit).

There are two main types of interfaces. The Network Inteface Controller (NIC) connects PMs to local rings and the Inter-Ring Interface (IRI) connects two rings. The NIC switches i) incoming packets to the input queue for the PM, ii) outgoing packets from the PM to the ring and iii) continuing packets from the input link to the output link. Each NIC has a bypass buffer capable of temporarily storing packets arriving from the previous ring interface while it is in the process of transmitting a packet from the local PM. The packets on the ring have priority over packets from the local PM.

The IRI controls the traffic between two rings. It is modelled as a 2x2 crossbar switch with input and output FIFO queues which can hold 10 flits each. The routing delay at a NIC is assumed to be 1 network cycle while it is assumed to be 2 network cycles at an IRI.

2.2 Simulator

We constructed a simulator that reflects the behavior of a system on a cycle-by-cycle basis, using the *smpl* simulation library [7]. The batch means method [7] of output analysis was used with the first batch discarded to account for initialization bias. The batch termination criterion was that each processor had to complete at least some minimum number of requests (in our simulations it is 200 requests per processor). The base simulator was validated against measurements taken from the Hector prototype [5]. It was then extended to model features not present in Hector, such as the insertion ring interface, flit level simulation, wormhole routing and flow control.

2.3 Benchmark Description

In order to evaluate the performance of the interconnection network under controlled conditions, we used synthetic benchmarks to drive our simulator. It was adopted from the Multiprocessor Memory Reference Pattern (M-MRP) address generator of Saavedra, et. al. [8], originally developed to measure real system performance. A M-MRP is a set of P Uniprocessor MRPs, one for each processor, each accessing memory in its own region. (The access regions of each processor may overlap.) Each M-MRP in our simulation is characterized by three attributes: 1) the number of processors, P, generating memory accesses, 2) the size of the memory region, R, accessed by each processor, 3) the cache miss rate, C, of each processor. By varying each of these attributes we can exercise the network in a specific and predictable way and can measure how the network responds under controlled conditions. Throughout our study, P is set to the number of processors in the system. Parameter R allows us to model different memory access patterns by varying it to control the degree of locality and thus the sharing between different processors, and indirectly R is used to control the amount of bisection traffic, i.e. traffic through the global ring. We assume that the memory region of a processor starts with its local memory module and that the sequence of memory references in a given region is uniformly distributed and independent across the region.

The micro benchmark generates a series of memory references at each processor as a result of cache misses. In our simulations, C is varied from 1/100 (1 miss in 100 cycle) to 1/20 (1 miss in 20 cycles), and R is varied from 1/P (the access region is contained entirely in local memory) to 1 (the access region covers all memories of the system). This range gives us sufficient variation in the amount of interconnect network traffic. In our simulations we do not model (dirty) cache line write backs, nor do we model invalidation packets.

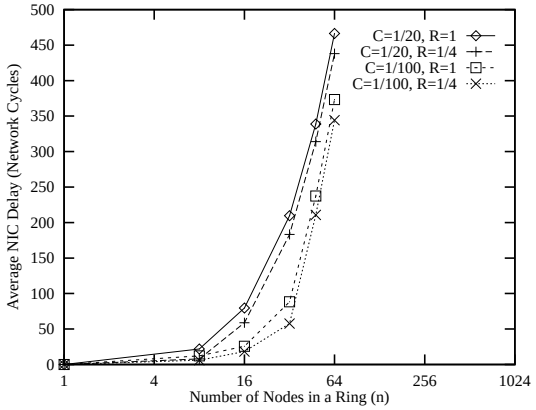

Figure 2: Single Ring Behavior: NIC Delay vs. Nodes

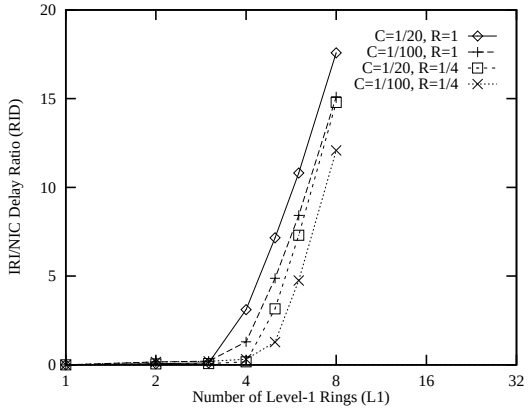

Figure 3: Two Level Ring Behavior: RID vs. L1 Rings

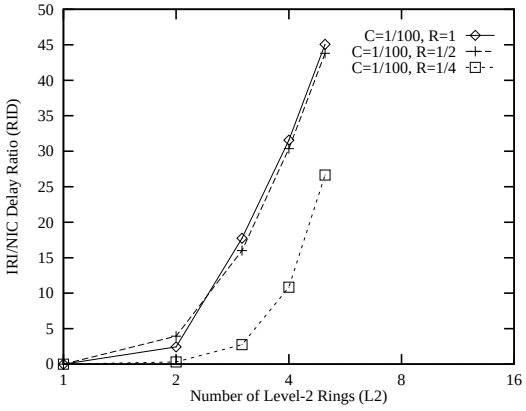

Figure 4: Three Level Ring Behavior: RID vs. L2 Rings

2.4 Measures of Network Performance

We use the following measures of performance in our study. The first two metrics gives us a direct measure of congestion in the network, the third metric measures the average latency in the network while the last one measures the severity of bisection bandwidth constraints:

Network Interface Controller (NIC) Delay is the total average time a packet spends in the Network Interface controllers. It is measured in terms of network cycles

Inter-Ring Interface (IRI) Delay is the total average time a packet spends in IRI controllers, also measured in network cycles.

Round-Trip Latency of a memory access is the elapsed time between when a request packet is generated and the corresponding response packet is received. This measure includes memory access time and is also measured in network cycles.

Ratio of Interface Delays (RID) is the ratio of IRI and NIC delays. This ratio measures the severity of bisection bandwidth constraints relative to local ring bandwidth constraints.

3.0 Hierarchical Ring Network Construction

In this section, we use a bottom-up approach to find hierarchial ring topologies that perform well. We start with a single local ring and determine the maximum number of nodes, n, it can accommodate while maintaining reasonable performance. We then add an extra level in the hierarchy and determine the maximum number of local rings L1, a second level ring can accommodate and so on.

Figure 2 shows the NIC delay plotted against the number of nodes in a single local ring for different values of R and C. This metric is chosen over average latency, since it is a direct measure of network congestion. The NIC delay for a single ring remains small and grows slowly up to 16 nodes for a high cache miss rate C of 1/20 and up to 32 nodes for a low cache miss rate C of 1/100. After this point, the NIC delay starts to grow rapidly. We also observe in Figure 2 that the performance of a single ring is less sensitive to the values of R and C if there are not more than 16 nodes. We therefore conclude that a local ring with n set to 16 should perform well for almost all access patterns and hence assume n to be fixed at 16 for the following discussions.

As a next step, we add a second level ring to the hierarchy. We would like to determine how many 16-node local rings, L1, can be sustained in a two-level hierarchy without major performance degradation. The results (not shown) indicate that with n is fixed at 16, NIC delays are small compared to IRI delays and are almost constant over the entire range of R and L1. The IRI delays are reasonable only for systems with $L1 \leq 4$.

Figure 3 plots the RID (the ratio of IRI and NIC delay) against L1. We choose this metric because the performance of the total system is dominated by the performance of global ring and RID measures how much worse the congestion at the global ring is compared to the local rings.There are two sets of curves in Figure 3: one for C=1/20 and another for C=1/100. In each of these sets

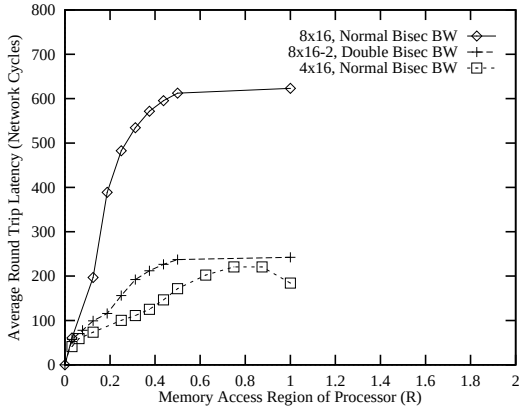

Figure 5: Two Level Ring Behavior: (Latency vs. R, C=1/100)

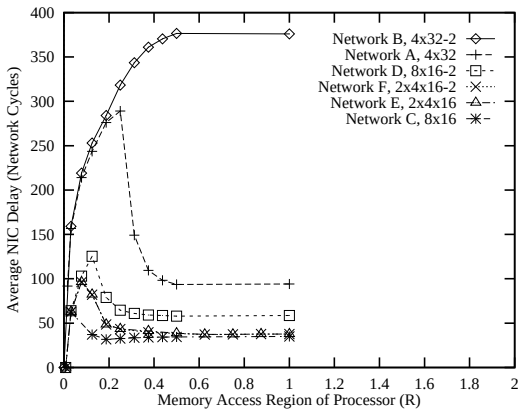

Figure 6: Hierarchical Ring Behavior(NIC Delay vs. R, C=1/20)

there are two curves: one for R = 1 (no memory locality) and one for R = 1/4 (high memory locality). Since it is apparent from Figure 3, that with $L1 \le 4$, the RID values tend to be smaller, and therefore we fix L1=4 for our larger systems. Again the performance of the system is less sensitive to values of R and C when up to four local rings are connected to a global ring.

Now, we introduce a third level in the hierarchy and proceed to determine how many Level-2 rings, L2, can be sustained in that level. Each L2 ring now consists of a second-level ring connected to 4 L1 rings of 16 nodes each, for a total of 64 nodes. Thus a L2 ring can be represented as 4x16. Figure 4 shows RID curves plotted against the number of L2 rings where it is apparent that RID values are smaller for $L2 \le 2$.

From the results obtained so far, we see a pattern that is easy to identify. With n=16, L1=4 and L2=2, we can no longer add a fourth level in the hierarchy. At this juncture, we observe that the bisection bandwidth constraint limits the size and scalability of the system for many access patterns. But a maximum size with 128 nodes three level of hierarchy could sustain most of the memory access patterns.

Now, instead of increasing the height of the network, we explore whether increasing the bisection bandwidth of a network would allow us to increase the system size without jeopardizing performance. The results are shown in Figure 5. It shows that a two-level, 64 processor system with a normal bisection bandwidth (4x16) has almost the same performance as a two-level, 128 processor system with double the bisection bandwidth (8x16-2). That is, we can increase the size of a hierarchical ring network, without increasing its height, by increasing its bisection bandwidth. In our model we double the bisection bandwidth by clocking the global ring at twice the speed. However, we will show in the next section that increasing the bisection bandwidth of a system, without increasing its size, may not always improve performance.

There are several options available for increasing the bisection bandwidth of a hierarchical ring system.

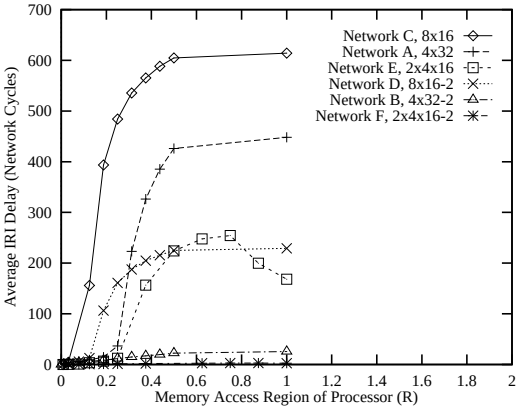

Figure 7: Hierarchical Ring Behavior (IRI Delay vs. R, C=1/20)

Above, we clocked the global ring twice as fast as the local ring. Alternatively, one could widen the channel width of the global ring, or have two global rings, that each connect to all next lower level rings. This is similar to the fat tree architecture [6], except that here we do not widen the channel width of the intermediate or lower-level rings.

4.0 Verification

In the previous section we attempted to compose as large a system as possible, using a bottom-up approach. To verify that the system we composed does indeed have the best topology for the given number of nodes, we compare its performance against a number of other topologies. We assume our goal is a system with 128 processors. The bottom-up approach resulted in a 3 level, 2x4x16 network. Alternatively, we could compose a 2-level, 8x16 network, if we double the bisection bandwidth. We compare these two networks with the four other topologies listed in Table 1.

Figures 6 and 7 show the NIC and IRI delay profiles for Networks A (4x32), B (4x32-2), C (8x16), D (8x16-2), E (2x4x16) and F (2x4x16-2) assuming a high cache miss rate of 1/20. Networks A, C and E are the same as

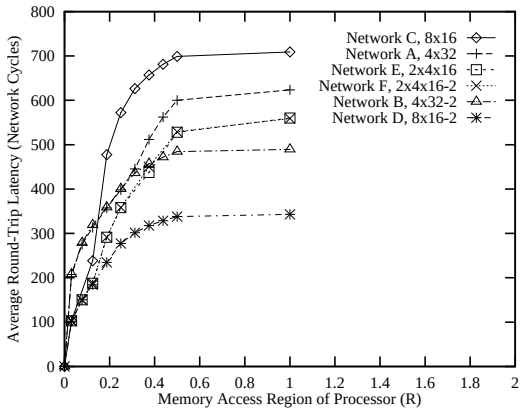

Figure 8: Hierarchical Ring Behavior (Latency vs. R, C=1/20)

networks B, D and F except that the latter have twice the bisection bandwidth. We observe in Figure 6 that for networks other than A and B, NIC delays are small for all R, except for small peaks around R=1/16. This behavior is quite consistent with our earlier results, which predict a low NIC delay for values of n less than or equal to 16. Here, the NIC delays increase until R=1/16, at which point appreciable traffic starts flowing through IRI interfaces causing severe congestion at those points. The high IRI delay caused by IRI congestion relieves congestion at the network interface controllers, causing a drop in NIC delay after R=1/16. For network A, the peak is much higher and occurs at R=1/32 due to 32-node rings. For network B, on the other hand, there is no peak, and the NIC delay remains large for all values of R. This observation indicates that increasing bisection bandwidth does not necessarily increase the performance of the system, but merely shifts the bottleneck from the global ring to the local rings.

In Figure 7, networks C and A have high IRI delays, networks E and D have moderate IRI delays and networks B and F have small IRI delays. Network E, which is our network of choice has lower IRI delays than any other network running the global ring at normal speed. (networks A and C). In network C, the IRI delay is much higher due to the fact that L1=8 is twice our recommended value. This causes severe IRI congestion. Network D, which is the same as network C, but with twice bisection bandwidth, shows much improved performance, where IRI delay is 60% lower. In this case, increasing bisection bandwidth improves performance considerably and reduces the congestion at the inter-ring interfaces. It is interesting to note that for both networks C and D, the NIC delays are small.

Finally, Figure 8 depicts the average latency against R, for all networks listed in Table 1, assuming a cache miss rate of C = 1/100. These curves gives us an idea of the overall impact of the local and bisection bandwidth constraints. We observe that network C, 8x16, performs

Table 1: Hierarchical Ring Configurations

NETWORK	TOPOLOGY
A	4x32, Normal Bisection BW
B	4x32-2, Double Bisection BW
C	8x16, Normal Bisection BW
D	8x16-2, Double Bisection BW
E	2x4x16, Normal Bisection BW
F	2x4x16-2, Double Bisection BW

worse than any of the other networks, while network D, 8x16-2, with twice the bisection bandwidth performs best. These latency curves supports our earlier hypothesis that network D with double bisection bandwidth is one of the best configurations for a system with 128 nodes, while network E is one of the best configuration if the global rings run at normal speed.

5.0 Concluding Remarks

In this paper, we used a bottom-up approach to determine how large hierarchical, ring based networks can become before the constant bisection bandwidth property of these networks begin to severely degrade performance. We showed that a system with 128 processors can sustain many memory access behaviors, but that larger systems perform adequately only if the memory access patterns exhibit good locality. Without locality, larger systems can be sustained, only if their bisection bandwidth is increased.

REFERENCES

[1] H. Burkhardt et al., *Overview of KSR1 Computer System*, Technical Report KSR-TR 9202001, (Feb 1992), Kendall Square Research

[2] W.J. Dally, “Performance Analysis of k-ary n-cube interconnection networks,” *IEEE Transaction on Computers*, (June,1990), pp. 775-785

[3] D.B. Gustavson, “The Scalable Coherent Interface and related standards projects,” *IEEE Micro*, (Feb 1992), pp. 10-22

[4] V.C. Hamacher and H. Jiang, “Comparison of Mesh and Hierarchical Networks for Multiprocessors,” *In Proceedings of the ICPP*, (August 1994), pp. 67-71

[5] M. Holliday, M. Stumm, “Performance Evaluation of Hierarchical Ring-Based Shared Memory Multiprocessors,” *IEEE Transactions on Computers*, (Jan 1994), pp. 52-67

[6] C.E. Leiserson, “Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing,” *IEEE Transactions on Computers*, (October, 1985), pp. 892-901

[7] M.H. MacDougall, *Simulating Computer Systems: Techniques and Tools*, MIT Press, 1987

[8] R.H. Saavedra et. al., “Characterizing the performance space of shared memory computers using Micro-Benchmarks,” *In Proceedings of Hot Interconnects*, (Aug 1993), pp. 3.3.1-3.3.5

[9] Z.G. Vranesic et. al., “Hector: A hierarchically structured shared-memory multiprocessor,” I*EEE Computer,* (January 1991), pp. 72-78

[10] Z.G. Vranesic et. al., *The NUMAchine Multiprocessor*, Technical Report CSRI-TR-324, (March 1995), Computer Science Research Institute, University of Toronto, Toronto