GigaRing Performance Analysis

Tomasz Kaczynski, Cray Research, Inc, Eagan, MN

ABSTRACT: *The GigaRing channel is a new, super-computer class system interconnect and IO channel developed by Cray Research based on the IEEE Scalable Coherent Interface(SCI) standard. The GigaRing channel has a ring based architecture, but can also function as a point-to-point link. The GigaRing channel has the potential of very high throughput combined with low latency while providing a high level of resiliency, flexibility and scalability. The performance of the GigaRing channel will be simulated and measured for a variety of configurations and workloads. The impact of factors such as block size, traffic distribution, number of channel nodes will be analyzed and discussed.*

1. Introduction

Cray Research has developed a new super-computer class system interconnect and IO channel called GigaRing. The architecture is based on the logical layer of the Scalable Coherent Interface (SCI) IEEE Standard, with significant reliability and performance enhancements.

The paper begins with a brief description of the GigaRing channel architecture in section 2. Channel operations are presented in section 3. A simulation model of the GigaRing architecture is described in section 4. Predicted GigaRing performance is presented in section 5, with some concluding remarks in section 6. Performance measurement of the GigaRing channel had to be postponed because there was no suitable GigaRing configuration available at the time this paper was written.

A comprehensive performance study of a single SCI ring is presented by Steven L. Scott et. al in [1]. An evolution and current status of the SCI standard is presented by David B. Gustafson et. al in [2].

2. GigaRing Channel Overview

The GigaRing channel consists of a *pair* of unidirectional, counter-rotating rings, which connect multiple *GigaRing Nodes* together with high speed, point-to-point links.

Figure 1 shows the basic organization of a GigaRing node which contains a *Client chip* and a *GigaRing Node Chip,* each implemented as a single ASIC. The GigaRing node chip has 32 bit wide input and output *links* and a bi-directional 64 bit wide *client port* interface to the client chip. For simplicity, the counter-rotating ring is not shown.

An example of the GigaRing channel used as a bi-directional, point-to-point channel is presented in Figure 3. This configuration is done by connecting two GigaRing nodes together. There are two physical paths travelling in each direction, but load balancing is performed automatically by the GigaRing node chips, creating the effect of a single link.

The architecture of the GigaRing channel encompasses three layers: a physical layer, a logical layer, and a protocol layer. The physical layer provides a physical communication medium. The logical layer

provides an efficient packet delivery service between clients over that medium. Finally, the protocol layer defines the standard for inter-client communication.

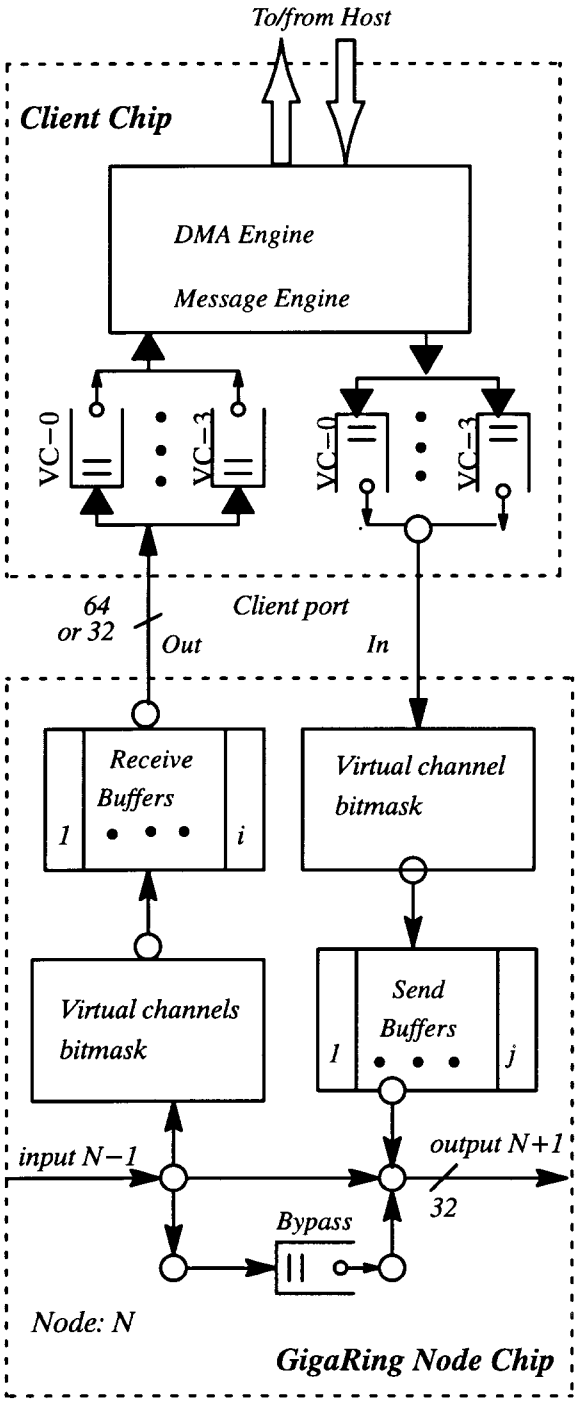

Figure 1. GigaRing node diagram. For simplicity the counter-rotating ring is not shown.

3. GigaRing Channel Operation

GigaRing implements a packet-based protocol. Nodes communicate by sending packets from *a source* node to *a target* node. Each packet consists of some number of contiguous *symbols* and contains a *header,* an optional *data payload*, and a trailing checksum. Symbols are 32-bits in length. When a node is not transmitting packet symbols on the link and there is no passing-through packet, it sends *idle* symbols, which contain housekeeping information for the ring. An example of a four-node GigaRing channel is presented in Figure 2.

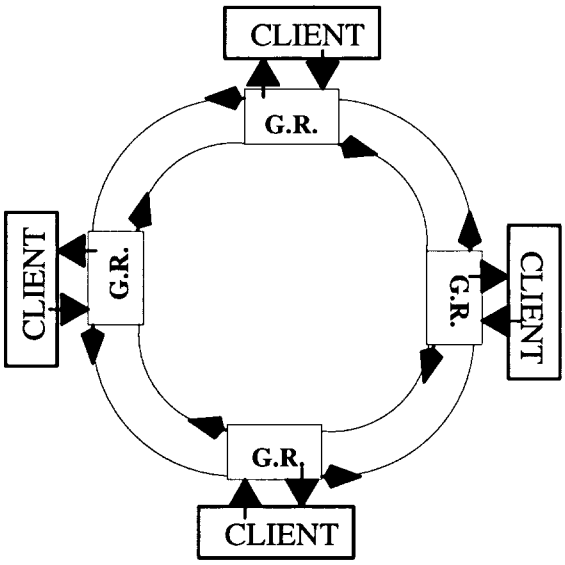

Figure 2. A four-node GigaRing channel.

When a client wants to send a packet, it places a *send* packet in the send buffer and then the packet is transmitted until it reaches its target node. At the target node, the send packet is stripped off the ring, placed into a receive buffer, and a small *echo* packet is routed the remainder of the way around the ring.

The echo packet informs the source node whether the send packet was successfully received at the target node or was *busied*. If the packet was accepted, then the source releases the send buffer occupied by the send packet. If the packet was busied, then the source node retransmits the packet.

A node is not permitted to transfer a packet from its send buffer unless its *bypass buffer* is empty. The symbols passing through the node that arrive during source packet transmission are directed to the bypass buffer. Since the bypass buffer can accommodate the maximum-sized packet, it can never overflow.

The GigaRing *arbitration protocol* operates independently on both rings of the channel. The protocol has two parts. Under contention, the *bandwidth alloca-*

tion protocol reserves an approximately equal portion of the ring bandwidth for each node. In the absence of contention, a single node can consume up to the full bandwidth of the ring. The *queue reservation* protocol prevents a packet from being "unlucky" and finding its target receive buffers full each time. Forward progress is therefore assured.

Although the channel protocol requires no back pressure on the links themselves, congested streams can waste channel bandwidth and send buffer space. The GigaRing channel provides two features to remedy this condition: multiple *virtual channels* to provide isolation between logically separate streams of packets and an *adaptive congestion control* mechanism which regulates data transfers to minimize retransmission traffic.

The GigaRing channel uses four virtual channels, corresponding roughly to read requests, read responses, write requests, and write responses. Separate flow control and buffering provided for each virtual channel allows traffic on different virtual channels to "slip" with respect to each other.

The adaptive congestion control mechanism works by limiting the number of *outstanding request* packets for each stream. When congestion occurs, the number of outstanding requests is reduced. As the congestion clears up, the number of outstanding packets is increased again.

The IO protocol supported by the GigaRing channel provides two capabilities to the clients: *peer-to-peer messaging* and *Direct Memory Access* - DMA. Peer-to-peer messages (up to 256 bytes of payload) are quite efficient, since they require no handshaking. DMA operations allow a "master" to directly access a "slave's" memory and generally they require some form of handshaking to establish the correct memory region within the slave. DMA transfers are performed through *ReadBlk* and *WriteBlk* operations

4. GigaRing simulation model

The GigaRing channel is a next generation interconnect technology based on the IEEE SCI standard. This is a new, complex technology and it was recognized that a GigaRing performance evaluation tool was needed. The combination of a huge performance space and the novelty of the GigaRing channel make it difficult to calculate or predict performance from previous experience. To fill the void, a detailed, parameter-driven simulation model of the GigaRing channel (GRM) was developed.

The GRM supports the GigaRing IO protocol described in section 3 and allows a user to specify channel, node, and workload parameters in the GRM *parameter file*. Using the GRM parameter file, one can specify arbitrary GigaRing channel configurations and generate a complex workload. The GRM parameter file also allows specification of a *warm-up* period. The statistics for the warm-up period are discarded, and the final results represent the steady state of the GigaRing channel.

Functionally, the GRM emulates the GigaRing node presented in Figure 1. GRM is composed of a number of independent *resource servers*. Each resource server handles a single resource of the GigaRing channel :Eg: buffers, links, port etc. GigaRing packets are represented in the model by an unshared datum which is passed between the resource servers.

Once the GR packet completes service at the current server, the unshared datum is updated and then passed over to another resource server. This process closely matches the way the hardware operates and allows relatively easy translation of channel specification into a workable model.

The model inputs describing the simulation are specified in the parameter file. For a parameter file, a GRM run produces an output file containing the throughput, latency and bandwidth utilization on a per channel, per ring, per node, and per link basis. In addition the GRM also produces a number of resource and traffic statistics.

While the GRM model was being developed an effort was made to validate the model as thoroughly as possible. The simulation results from the GigaRing model were validated against the well established model of the SCI ring described in [1], against mathematical calculations done by GigaRing designers, and against a low level Verilog simulator. In all instances data from GRM were found to be in good agreement.

5. GigaRing performance

The GRM was applied to simulate the impact of transfer size on the peak payload characteristic of the GigaRing channel. In this prediction scenario the channel was configured with 2 nodes (see Figure 3 below) and the simulations were design to stress the channel itself. Concurrent data transfers were applied.

The point-to-point connection represented in Figure 3 is the simplest way to configure a GigaRing channel. The advantage of this configuration is dedication of the channel bandwidth to the two connected nodes (the bandwidth is not shared with any other nodes).The channel characteristic was simulated for Read Block transfers with two traffic scenarios:

- *half duplex* i.e. only one node is transmitting data

•*full duplex* i.e. both nodes are transmitting data

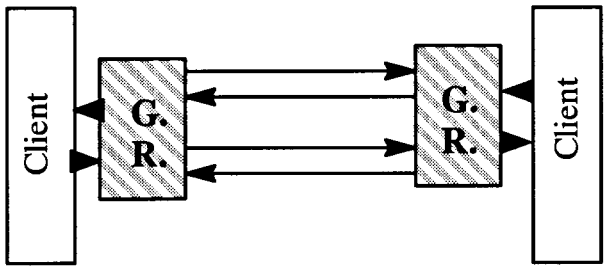

Figure 3. GigaRing used as point-to-point channel

The half duplex peak payload throughput characteristic is presented in Figure 4 and full duplex characteristic in Figure 5. The effect of transfer size on channel latency for half duplex and full duplex traffic is presented in Figure 6.

The GigaRing peak data payload characteristic for a half duplex traffic (see Figure 5) saturates at 916 MB/s for block size of about a kiloword. For small block sizes (less than or equal to 32 words), GigaRing traffic is dominated by the IO protocol overhead. For block sizes greater than 32 words there is more than one data packet per transferred block and the significance of the IO protocol overhead diminishes quickly.

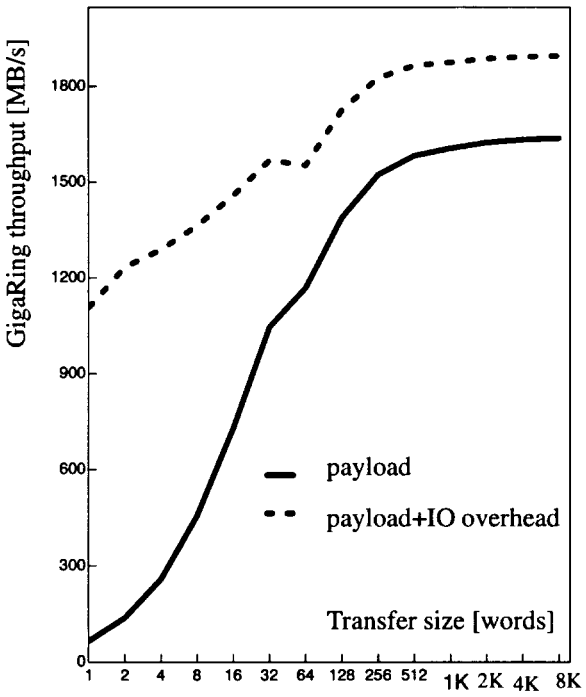

Figure 5. Effect of transfer size on peak throughput, full duplex, RdBlk transfer.

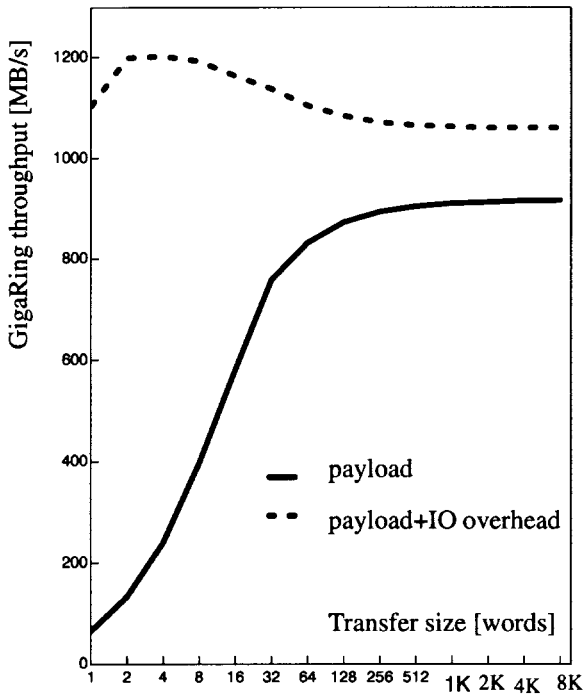

Figure 4. Effect of transfer size on peak throughput, half duplex, RdBlk transfer.

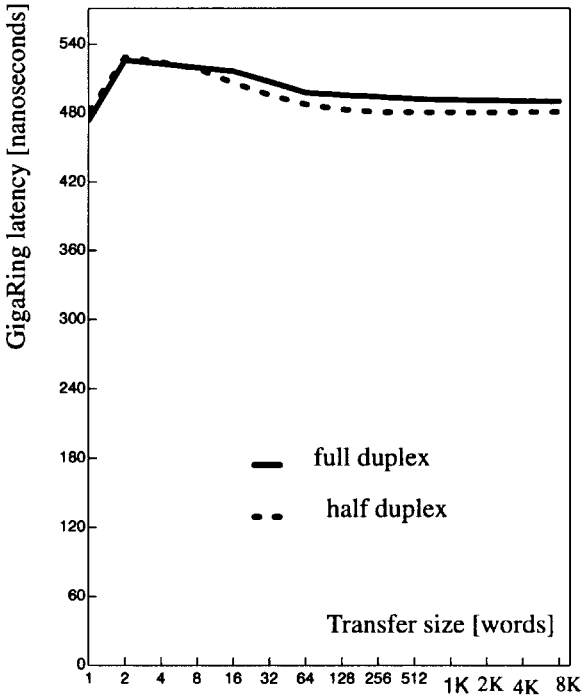

Figure 6. Effect of transfer size on channel latency.

The full duplex peak payload throughput characteristic (Figure 5) saturates at a level of 1,638 MB/sec (2x819 MB/sec) for a block size of 2 kilowords.

The GigaRing channel latency (time between sending a data packet and receiving and echo packet) is shown in Figure 6. The unloaded channel latency was 462 nanoseconds, when the inter-node wire length was set at zero meters. The unloaded channel latency for the configuration presented in Figure 4 should be increased by 5 nanoseconds for each meter of wire between nodes.

6. Conclusion

The GigaRing channel features high bandwidth, low latency performance with peak payload bandwidth of:

- half duplex client to client DMA = 916 MB/sec
- full duplex client to client DMA = 2 x 819 MB/sec

Given the high performance of the GigaRing channel, hardware latencies for short transfers are quite small. The overall system performance for shorter transfers therefore will likely be bound by the software latencies. To achieve good performance over the GigaRing channel, one should use a combination of long transfers, low software overhead and/or concurrent transfers.

The experience with the simulation model of the GigaRing channel showed usefulness of this tool in evaluating the GigaRing design alternatives as well in investigating the performance of different channel configurations.

7. Acknowledgments

I'd like to thank to Steve Scott, Bob Halford, Duane Boetcher, Gary Schwoerer, Daniel Kunkel, Jim Bedell and John Melom for their patience and willingness to discuss the GigaRing issues.

8. References

[1] Steven L. Scott, James R. Goodman and Mary K. vernon. Performance of the SCI Ring. 1992 ACM 0−89791−509−7/92/0005/0403.

[2] David B. Gustavson, Quiand Li. Local-Area Multiprocessor: the Scalable Coherent Interface, SCIzzL, Santa Clara University.