Performance Evaluation of a Multicast Mechanism For a Massively Parallel Processor JUMP-1

Noriaki Suzuki, Hideharu Amano,
Tomonori Tamura, Yasunori Osana
Department of Computer Science
Keio University
Yokohama, Japan

Katsunobu Nishimura
Faculty of Commerce and Economics
Chiba university of Commerce
Chiba, Japan

Abstract

A massively parallel processor JUMP-1 has been developed for building an efficient cache coherent distributed shared memory on a large system with more than 1000 processors. In this paper, we evaluate a multicast mechanism and generation/collection of acknowledge packets in JUMP-1.

In a router of the network RDT(Recursive Diagonal Torus) and a distributed shared memory management processor MBP-light, hardware mechanisms for packet multicasting are equipped. Using a real machine, the performance improvement compared with software is measured.

Keywords: Multiprocessor, CC-NUMA, Performance Evaluation, Interconnection Network, JUMP-1

1 Introduction

A Cache Coherent Non-Uniform Memory Access machine (CC-NUMA) is one of hopeful candidates for future common high performance machines. Unlike bus-connected multiprocessors, the system performance can be enhanced scalably as to the number of processors. Moreover, parallel programs developed in small multiprocessors can be ported easily.

Presently, commercial CC-NUMA machines, such as SGI Origin 2000[1] or Sequent NUMA-Q[2], has been developed. However, when thousands of processors are connected, a large amount of memory and hardware are required to manage the DSM.

JUMP-1 is a prototype of a massively parallel processor with cache coherent DSM developed by collaboration of seven Japanese universities[3]. The major goal of this project is to establish techniques required to build an efficient DSM on a massively parallel processor. In order to satisfy both high degree of performance and flexibility, JUMP-1 has several distinctive structures. Interconnection Network called RDT[4] includes both torus and a kind of fat tree structure with recursively overlayed two-dimensional square diagonal tori. A dedicated processor called MBP(Memory Based Processor)-light[7] is proposed to manage the DSM of JUMP-1. MBP-light consists of a simple dedicated core processor and hardwired controllers which handle memory systems, bus and network packets.

In this paper, we evaluate a multicast mechanism and generation/collection of acknowledge packets in JUMP-1. In a router of the network RDT and a distributed shared memory management processor MBP-light, hardware mechanisms for packet multicasting are equipped. Using a real machine, the performance improvement compared with software is measured.

2 The Structure of JUMP-1

As shown in Figure 1, JUMP-1 consists of 256 clusters connected each other with a network RDT. Each cluster provides a high speed point to point I/O network connected with disks and high-definition video devices.

Each cluster is a bus-connected multiprocessor, as shown in Figure 2, including four RISC processors (SuperSPARC+), MBP-light which is directly connected to a cluster memory, and RDT router chip for interconnection network[5]. MBP-light, the heart of JUMP-1 cluster, is a custom designed processor which manages DSM, synchronization, and packet handling.

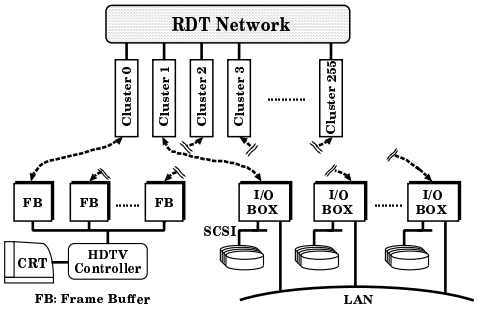

Figure 1: The Structure of JUMP-1

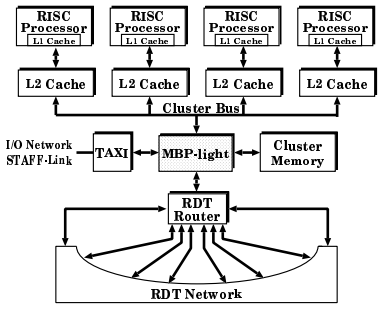

Figure 2: The Structure of JUMP-1 Cluster

2.1 Interconnection Network - RDT

The RDT[4] is a network consisting of recursively overlayed two-dimensional square diagonal tori. In order to reduce the diameter, bypass links are provided in the diagonal direction. When four links are added between a node (x, y) and nodes $(x \pm n, y \pm n)$ (n: *cardinal number*) respectively, additional links result in a new torus-like network. A new torus-like network is formed at an angle of 45 degrees to the original torus, and the grid size is $\sqrt{2}n$ times that of the original torus. We call this new torus-like network the rank-1 torus. On the rank-1 torus, we can form another torus-like network (rank-2 torus) by providing additional links in the same manner. Figure 3 shows rank-1 and rank-2 tori when n is 2. The RDT consists of such recursively formed tori.

Recursive Diagonal Torus RDT(n,R,m) can be defined as a class of networks in which each node has links to form base (rank-0) torus and m upper tori (the maximum rank is R) with cardinal number n. Note that, each node can select different rank of upper tori from others.

A large degree makes implementation difficult. JUMP-1 must be scalable to the system with ten thousand nodes, therefore m is to be

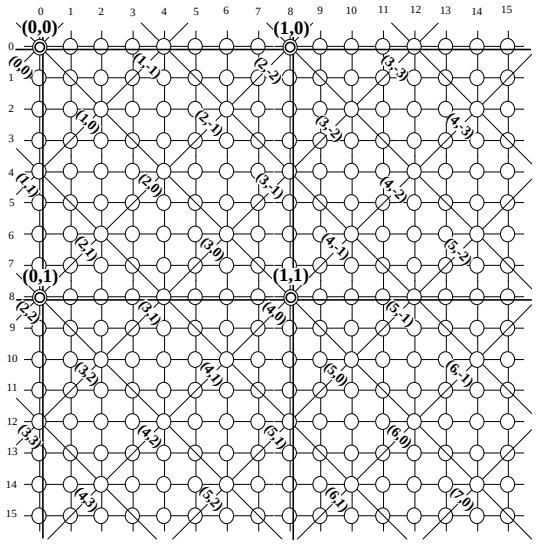

Figure 3: Upper rank tori

1 (degree = 8). For this number of nodes, the rank of the most upper torus is 4. Thus, the RDT(2,4,1) is adopted here.

In the RDT, each node can select different rank tori from others. Thus, the structure of the RDT(2,4,1) also varies with the rank of tori which are assigned to each node. This assignment is called the *torus assignment*.

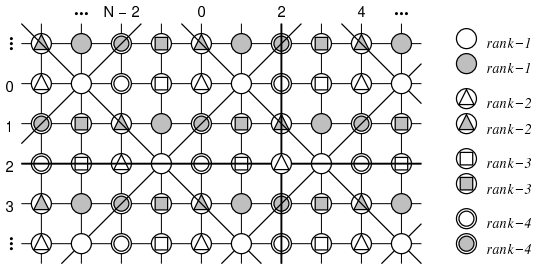

Figure 4: Torus assignment used in the JUMP-1

In this assignment, a node has eight links, four for the base (rank-0) torus and four for one of the upper rank (1-4) tori (most links to upper rank tori are omitted in Figure 4). Note that all nodes have neighboring nodes which is connected three other ranks except each own rank. Therefore, the torus of any rank can be used after at most single message transfer between neighboring nodes. This property re-

duces the diameter and average distance between nodes.

2.2 RDT Router Chip

2.2.1 Structure of the router chip

The structure of RDT router chip is shown in Figure 5. The core of the chip is a 10×11 crossbar which exchanges packets from/to ten 18-bit wide links, that is, four for the rank-0 torus, four for the upper rank torus, and two for the MBPs which manage the distributed shared memory of JUMP-1. In JUMP-1, two RDT router chips are used in the bit-sliced mode to form 36-bit width for each link.

All packets are transferred between router chips synchronized by a unique 50MHz clock. In order to maximize the utilization of a link, packets are bi-directionally transferred. The maximum packet length is 16 flits (36-bit width 16-flit length) which can carry a cache line. 3-flit header which carries the bit-map of RHBD(Reduced Hierarchical Bitmap Directory)[6] is attached to every packet, and the length of the body is variable.

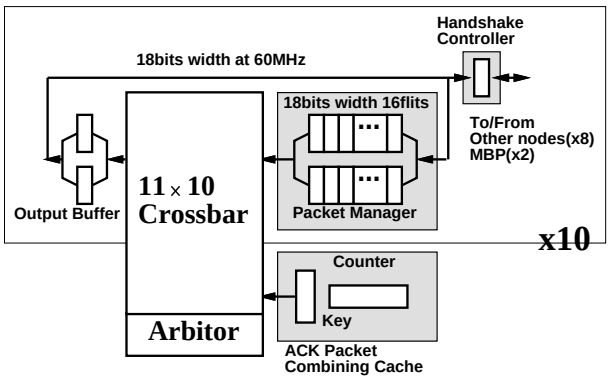

Figure 5: The structure of the RDT router

2.2.2 RHBD on RDT

Since the RDT includes a fat tree of tori with multiple root nodes, the RHBD can be implemented in the distributed manner without causing congestion around the root node. The pattern of message transfers for emulating a fat tree is shown in Figure 6.

Two steps are required: (1) each node transfers a message to four neighbors, (2) a neighbor (the south in this figure) transfers the message to three neighbors. Thus, if all nodes with

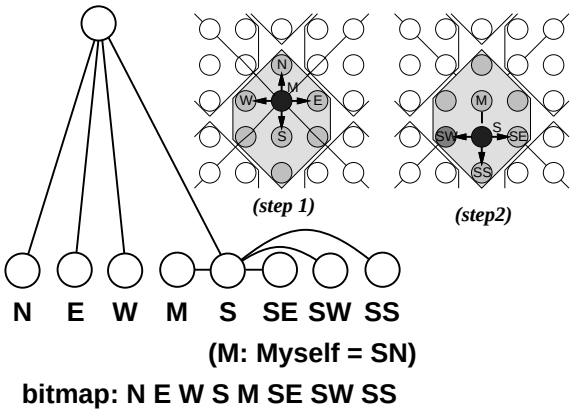

Figure 6: 8-ary tree and bit-map pattern for multicast on RDT

rank-i tori execute this pattern, the message is transferred to all nodes with rank-(i-1) tori. By repeating this data transfer from the maximum rank to the rank-0, 8-ary tree is formed on the RDT. In this case, a rank in the RDT directly corresponds to the level of the tree. Moreover, in the RDT, the upper rank torus can be used within a step of message routing. Thus, a message can be directly transferred from the sender node to the root node without using the tree structure. Figure 6 also shows the 8-ary tree relevant to this bit-map pattern for the hierarchical bit-map directory scheme.

In the RDT, nodes which receive the message through the tree whose root rank is 'i' are located around the source node. For larger 'i', the number of such nodes becomes large, thus the area in which a message is multicast becomes wider. We call such an area "territory" of a multicast. Figure 7 shows territories of a multicast from rank-0 and rank-1. Since the territory is always formed around a source node, message multicast to local nodes is performed from a lower rank (thus, with only a small territory).

2.2.3 Method of multicast

In this router chip, the asynchronous wormhole routing is adopted to cope with the frequent multicasting. Although a packet can be forwarded to the buffer in the next node while receiving the packet like a usual wormhole routing, a buffer which can hold the maximum sized packet is provided for each virtual channel. When the target buffer is occupied,

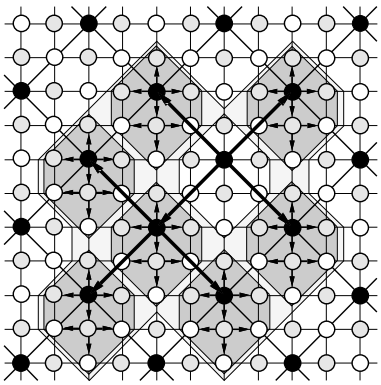

Figure 7: Territory of a multicast

the entire packet can be stored in the buffer inside the chip. This technique is not the virtual cut through since infinite number of buffers are not provided. However, communication to the buffer outside the chip which may cause the severe performance degradation can be avoided.

The simplest way for multicasting is to wait until all required opposite buffers become empty and multicast at a time. However, the opportunities of multicast are badly reduced when the number of the multicast destinations is increased. In the RDT router chip, each buffer provides a bit-map corresponds to required destinations of a packet. The packet is sent whenever the empty opposite buffers are found, and then the corresponding bits of the multicast bit-map are reset.

2.2.4 Chip Implementation

RDT router chip is implemented on HITACHI's $0.5\mu m$ BiCMOS gate array HG22S125. When designing the router chip, crossbar and arbiter that require quick operations are designed by the schematic editor and the other sections are described in VHDL. The router chip is implemented in a PGA package and provides 299 pins. Maximum clock of a router is 60 MHz and area utilization is 63 percent. The specification of RDT router chip is shown in Table 1.

2.3 MBP-light

The structure of MBP-light is shown in Figure 8. MBP-light is consists of three modules: RDT interface to treat network packets, MMC (Main Memory Controller) to control cluster memory and cluster bus, and MBP Core which is the core processor. RDT Interface and MMC provide their own hardware mechanisms, and work independently from MBP Core.

Table 1: The specification of RDT router chip

Maximum clock (MHz)	60
Consuming power (W)	19.4
Area utilization (%)	63
The number of pins	299
Package	PGA

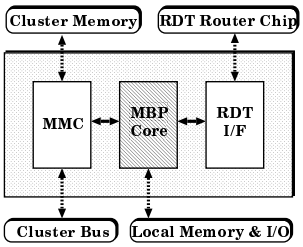

Figure 8: The Structure of MBP-light

2.3.1 MBP Core

MBP Core consists of a pipeline with four stages treating 21-bit instructions and 16-bit data. 21-bit x 64K local memory which stores instructions and local data is connected.

Since jobs which must be quickly processed are mostly managed by the hardware mechanisms in RDT Interface and MMC, MBP Core only processes a complicated part of the DSM Protocol. It mainly decodes a packet, accesses a table, transforms the address, and generates the packet to send somewhere.

2.3.2 RDT Interface

RDT Interface is directly connected with RDT router chip and manages network packet transfer.

For avoiding network congestion, packets must be multicast in the network (one-to-one transfer is so inefficient). Since RDT network used in JUMP-1 provides the hierarchical multicast mechanism, it can be done without network congestion[6]. For a protocol processor, a fast generation and collection of acknowledgment packets are essential. RDT Interface

provides two dedicated mechanisms: Ack Generator and Ack Collector for this purpose, as shown in Figure 9.

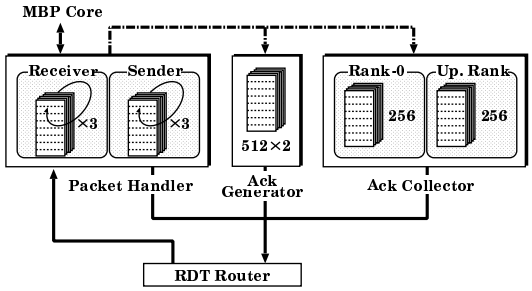

Figure 9: The Structure of RDT Interface

Ack Generator provides two cache systems called Net Cache and Ackmap Cache. Both caches are accessed when a packet with coherent message is received. Net Cache, a direct mapped cache with 512 entries, is accessed by the address of the DSM in the packet header. It stores the information whether the accessed line is cached in the cluster (in L2 or L3 cache) or not. At the same time, Ackmap Cache is accessed by the source cluster number of the receiving packet, and the bitmap which shows the returning path is obtained. Using the above information, an acknowledgment packet when the accessed data is cached, or a not-acknowledgment packet when the accessed data is not cached is automatically generated.

On the other hand, when an acknowledgment packet is received, another cache system called Ack Cache in Ack Collector is accessed by the key in the packets. Ack Cache is a direct map cache which provides 128 entries for each hierarchy of the embedded tree in the RDT network. The number of packets which must be collected is registered in the cache entry, and the number is decremented when a packet arrives. When the number becomes zero, another acknowledgment packet for the upper hierarchy is generated, or MBP Core is interrupted.

In both cache systems, if a miss occurs, the program of the MBP Core will be interrupted for replacing or generating the entry. In this case, the performance is much degraded.

2.3.3 Chip Design

The MBP-light is implemented on TOSHIBA's $0.4\mu m$ CMOS 3-metal embedded array TC203E340. In order to cope with a large number of pins (352 pins), TBGA (Tape Ball Grid Array) package is used. The design of MBP-light is described in VHDL, synthesized with Mentor Autologic-II, and verified with TOSHIBA's VLCAD. The maximum clock of MBP-light is 50 MHz. The number of gates for random logics is 106,905 and internal memory for core processor is 44,848 bits. The specification of MBP-light is shown in Table 2.

Table 2: The specification of MBP-light

Maximum clock (MHz)	50
Consuming power (W)	3.1
Area utilization (%)	38.3
The number of pins	352

3 Performance Evaluation

We evaluate the multicast mechanism and generation/collection of acknowledge packets by using a real machine. Now, a system with 64 processors (16 clusters) is available as shown in the Figure 10.

Figure 10: Prototype JUMP-1 system

3.1 Packet Multicasting

First, we evaluate the multicast mechanism. Clock cycle time for sending a packet to multiple nodes is measured and compared with unicasting.

Figure 11 shows the result of evaluation. When the packet is multicast, the bitmap of

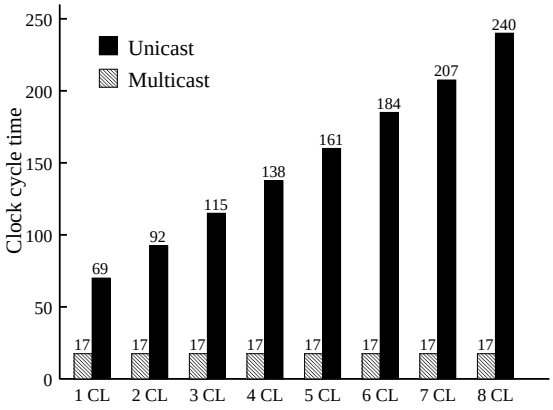

Figure 11: Clock cycle time for packet sending

the RHBD is attached into the header of multicast packet directly. On the other hand, when packets are sent by unicasting, the header of each packet must set by examining the bitmap of the directory. Therefore, even if a packet sent to one node, unicasting requires 69 cycles while multicasting requires 17 cycles. Furthermore, multicasting requires only 17 cycles even if packet sent to multiple nodes, while cycles for unicasting increase as a number of receiver nodes.

3.2 Generation of Acknowledge Packet

Next, we evaluate the automatic generation mechanism of the acknowledge packet. The turn around clock cycle time is shown in Table 3. The cycle time is 51, while management software processed by MBP-light's core processor requires 91 clocks. At 50MHz clocks, the hardware management can save 40 cycles(800 ns) compared with the software.

Table 3: Clock cycle time for generating acknowledge packet

With hardware mechanisms	51 cycles
Software management	91 cycles

3.3 Collection of Acknowledge Packets Managed by MBP-light

Next, we evaluate the automatic collection mechanism of acknowledge packets in the MBP-light. A multicast packet is sent from a sender node to some receiver nodes, and after the packet arrives, the hardware acknowledge generation mechanism in receiver nodes return an acknowledge packet back to the sender node. Then, the sender node collects these acknowledge packets by using the hardware collection mechanism. Clocks required for the above steps are measured and shown in Figure 12. When a multicast packet is sent to seven nodes, the hardware management can save 132 cycles(2640 ns at 50 MHz clocks) compared with the software.

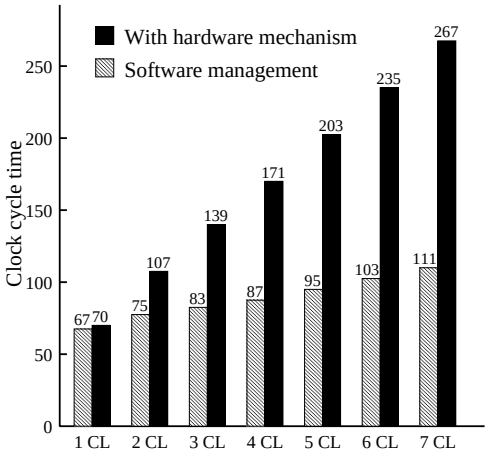

Figure 12: Clock cycle time for collecting acknowledge packets

3.4 Execution of LU decomposition

Now, DSM of JUMP-1 is under debugging, and works only in limited conditions. Thus, we evaluate the performance improvement with a small parallel benchmark program: LU decomposition. Here, the program is executed in 4 clusters, and 1 processor per cluster is used. A small matrix with 16×16 is used for generating flexible communication.

Table 4 shows a number of multicast packets sent from a home cluster (invalidate request/read request) and a number of multicast packets or acknowledge packets received in the home cluster. Execution clock cycle time is also shown.

The result shows that unicasting must process the packets to send 1.2 times as many as the packets sent by multicasting. Software management of collecting the acknowledge packets must process received packets 1.2

Table 4: result of LU

send policy / ack collection policy		multicast automatic	unicast software
send	Mult. packet (Inv. req)	57	92
	Mult. packet (Read req)	165	172
receive	Mult. packet	169	175
	Ack. packet	57	92
clock cycle time		2640023	3750258

times as many as the packets managed with hardware mechanisms. By using a multicast mechanism and automatic collection mechanisms, hardware management can save about 1110000 cycles compared with software.

4 Conclusion

A multicast mechanism for a massively parallel processor JUMP-1 is introduced, and its performance is evaluated by using a real machine.

In the RDT router chip and a distributed shared memory management processor MBP-light, hardware mechanisms for packet multicasting and generation/collection of acknowledge packets are equipped. By using those mechanisms, required clock cycle time for sending the same packet to some nodes are reduced and collection of acknowledge packets is processed efficiently. Then, hardware mechanisms reduce the execution time for LU decomposition.

The results demonstrate that a multicast mechanism works effectively to improve the performance.

Now, DSM on JUMP-1 is under debugging, and we will evaluate the performance improvement with practical parallel benchmarks.

References

[1] J. Laudon and D. Lenoski,"The SGI Origin 2000: A CC-NUMA Highly Scalable Server", Proceedings of the 24th Annual International Symposium on Computer Architecture, pp.241-251, 1997

[2] T. Lovett and R. Clapp,"STiNG: A CC-NUMA Computer System for the Commercial Marketplace", Proceedings of the 23rd Annual International Symposium on Computer Architecture, pp.308-317, 1996

[3] K. Hiraki et. al., "Overview of the JUMP-1, an MPP Prototype for General-Purpose Parallel Computations", IEEE International Symposium on Parallel Architectures, Algorithms and Networks, pp.427-434, 1994

[4] Y. L. Yang et. al., "Recursive Diagonal Torus: An interconnection network for massively parallel computers", IEEE symposium on Parallel and Distributed Processing, pp.591-594, 1993

[5] H. Nishi et. al., "Router Chip: A versatile router for supporting a distributed shared memory", The IEICE transactions on Information and Systems, 1997

[6] Tomohiro Kudoh et al., "Hierarchical Bit-map Directory Schemes on the RDT Interconnection Network for a Massively Parallel Processor JUMP-1", Proceedings of the International Conference on Parallel Processing, volume I, pp.186-193, 1995

[7] I. Hiroaki et al., "MBP-light: A Processor for Management of Distributed Shared Memory", International Conference on ASIC, pp.199-202, 1998