# An Overview of Reflective Memory Systems

Milan Jovanovic, Veljko Milutinovic

University of Belgrade

*E-mail: {ejovanom, vm}@ etf.bg.ac.yu*

**Abstract**

*This is a survey of reflective memory systems - an emerging approach to distributed shared memory which has existed for decades now. Reflective memory systems are qualified by a high-speed memory-mapped communication medium that assures coherent replication of data which are kept in local memory; this implies a low and constant memory access time. In addition, inherent overlapping of computation with communication, fault tolerance, and support of open systems and off-the-shelf components make these systems useful for real-time and many other applications. Commercially available reflective memory systems (Architecture Technology Corporation, DEC, Encore, Modcomp, Systran, VMIC), as well as academic research projects (Florida, Princeton, Stony Brook, Tokyo, Belgrade), are described in a uniform manner.*

## 1. INTRODUCTION

The communication latency of conventional message-passing multicomputers, especially traditional networks, is usually too large to maintain good processor efficiency because operating system overhead takes too much time. Also, too much processor time is consumed with the layered protocol software. The communication latency is hard to predict, which is especially important for real-time applications. As the number and the speed of processors increase, the above problems become more acute. Therefore, the design of an efficient interconnection policy is crucial.

Reasonable solutions to these problems represent systems based on non-demand updates of remote shared memory copies, which are the subject of this paper. There are differences in designation of these systems. Some authors call them mirror memory systems [e.g., Witt89] or replicated memory systems [e.g., Oguc95]. However, we refer to them as reflective memory (RM) systems [e.g., Enco90], which is more common.

These systems belong to the class of distributed shared memory (DSM) systems [Nitz91, Prot98]. Each processor includes a dual-ported local physical memory, a part of which is configured as logically shared. All these physically distributed logically shared parts of memory are mapped into a global shared address space and make the reflective memory. The main idea of the RM concept is to keep an accurate copy of a shared data item in a local memory of each processor, if there is a possibility that it would be used later. Upon each write to a shared memory location, the network interface (without explicit demand) updates all copies around the system. Therefore, the shared reads are always satisfied from the local memory; only shared writes are propagated through the system.

Presented RM systems are introduced in Table 1. It includes commercially available systems, as well as university research projects. Encore has a few bus-based systems [Enco90, Gert93] suitable for different purposes. Modcomp [Furh97], Systran [Syst95], and VMIC [Vmic95] have commercially available systems dedicated to the real-time market. The RSM is projected for the case where a small number of nodes are distributed over a wide area [Oguc95]. Having a cluster in mind,

Architecture Technology Corporation (ATC) has proposed a ring-based approach [Rama95], while DEC has designed a Memory Channel [Gill96] based on a crossbar. The MERLIN [Witt89] and its successor SESAME [Witt92] are intended to scale up efficiently to thousands of processors in a heterogeneous environment. The SHRIMP [Blum94] is one more scalable multicomputer based on a mesh.

| System | Academic/ Industrial | University/ Company | Year Introduced | Interconnection Network | URL |
|---|---|---|---|---|---|
| RMS | I | Encore | 1990 | bus | http://www.encore.com |
| RMS for PC | A+I | Belgrade/Encore | 1993 | bus | http://galeb.etf.bg.ac.yu/~dsm |
| RM/MC | I | Encore | 1993 | bus | http://www.encore.com |
| RM/MC++ | A+I | Belgrade/Encore | 1994 | bus | http://galeb.etf.bg.ac.yu/~dsm |
| LAM | I+A | Florida/Encore | 1996 | bus | http://www.encore.com |
| MMM | I | Modcomp | 1993 | bus | http://www.modcomp.com |
| RSM | A | Tokyo | 1994 | bus | http://www.sail.t.u-tokyo.ac.yp/~oguchi |
| MC | I | DEC | 1996 | crossbar | http://www.digital.com |
| NSM | I | AT C | 1994 | ring | http://www.atcorp.com/research/cluster |
| SCRAMNet | I | Systran | 1991 | ring | http://www.systran.com |
| VMIC | I | VMIC | 1995 | ring | http://www.vmic.com |
| Sesame | A | Stony Brook | 1991 | mesh | ftp://ftp.cs.sunysb.edu/pub/techreports/wittie |
| SHRIMP | A | Princeton | 1994 | mesh | http://www.cs.princeton.edu/shrimp |

**Table 1.** Introduction to the presented RM systems

***Description.*** *A- Academic; I-Industrial.*

# 2. THE GOALS OF THIS PAPER

The major purposes of this survey are as follows. First, the goal is to explain the RM concept. Second, the goal is to describe what has been done in this area. This is achieved by surveying a representative set of existing RM systems in a uniform manner emphasizing differences, advances, and limitations. Third, the goal is to predict the trends of future research and development of these systems. The importance of the aforementioned goals stands behind the emerging popularity of systems based on the RM concept.

The presentation of RM systems follows a uniform organizational structure: origin of the system, the most indicative picture, essence of the system (including the basic idea, the architecture and organization, functional description, interconnection technology, topology, memory consistency model (MCM), level of updating, level of sharing, caching of shared regions), typical applications, as well as advantages and disadvantages compared to other RM systems.

Performance of presented systems is not considered here, because these systems are designed having different purposes in mind and their performances are often not comparable. Also, there are systems which have been implemented in an older technology and their performance is not what it could be if implemented in present-day technology. Performance analysis of RM systems based on simulation modeling and "common denominator technology" is beyond the scope of this paper.

# 3. REFLECTIVE MEMORY SYSTEMS

The RM systems can be very different. The type of interconnection network used for RM updates is unimportant and can be: bus, bus hierarchy, ring, mesh, crossbar, etc. The granularity of sharing can be: word (rarely), page (usual), or segment (promising). The level of updating can be word or block. The mapping of shared memory regions can be dynamic or static. The MCM can be strict, sequential, processor, release (with all its variations), or entry (with all its variations). Caching of RM is rarely supported; however, there are systems with this capability.

Major advantages compared to other DSM systems are as follows. The computation in RM systems is usually overlapped with the communication. The memory access time of RM is usually constant (and therefore deterministic) and relatively low (which is important for real-time and many other applications), because shared data are almost always in local memory. A multiple_reader/multiple_writer DSM management algorithm is supported. Due to the inherent replication, these systems are good for fault tolerance. Also, these systems usually take advantage of open systems and off-the-shelf components. Reflective memory systems are simpler, so they have been commercially implemented for decades now.

Of course, there are disadvantages. For applications characterized with longer sequences of writes to the same word by a particular processor some unnecessary update traffic may be produced. Generally, data transfers are often performed because each shared write causes reflection, so the interconnection medium usually represents a bottleneck. Furthermore, concerning the interconnection medium, one-to-all broadcast communication has to be supported. The RM is usually not implemented on the same board with the processor, so its access time is slightly longer than that of the local memory implemented on the same board. In order to keep the RM consistent, an explicit synchronization of processes that write into the same shared memory location is needed.

## 3.1. Encore RM/MC System

The RMS (Reflective Memory System) represents the Encore Computer Corporation's implementation of RM dedicated to the real-time market. This bus-based system supports word updates only. It has been derived from the first RM system designed and patented by Gould Electronics in 1985 (Encore acquired Gould in 1989). The RMS system was upgraded to support block updates, making RM/MC (Reflective Memory/Memory Channel) system [Gert93]. Because the RM/MC system is the state of the art, only it will be described here.

**Essence.** The system consists of up to eight processing nodes (Figure 1) connected by means of the multiplexed synchronous 64 bits wide RM/MC bus that supports both single word and block update transfers. In order to expand the connection bandwidth, more parallel connected RM/MC buses can be used. The arbitration on the RM/MC bus is centralized and a round robin synchronous arbitration algorithm is used. Supported distances are: 80 feet with copper cables, 240 feet with coaxial cables, and 15000 feet with a fiber optic cable.

The challenge was to use commodity components in open system environments. To accomplish this goal the node is divided into two parts: TMI (Transition Module Interface) board which implements the interface to the RM/MC bus, while HPI (Host Port Interface) board provides the interface to the host system bus. There are realizations for the VME bus, the EISA bus, and the PCI bus.

Local memory pages can be configured as reflective (shared) or private (non-shared) by means of translation windows on transmit and receive sides. These windows are used for determination

whether data from a particular page have to be transmitted to and/or received from the RM/MC bus, as well as for address mapping (copies of a shared page could be at different addresses on different nodes). Receive and transmit FIFO buffers are provided to make the transfers to/from the RM/MC bus going asynchronously from/to the host.

**Applications.** The RMS supports numerous real-time applications such as vehicle simulation, telemetry, instrumentation, and nuclear power plant simulation and control. The RM/MC system was initially designed to satisfy the OLTP (On-Line Transactions Processing) application needs. Beside that, projects employing this system have included a mainframe-class data storage, a data base distributed lock manager, and decision support applications.

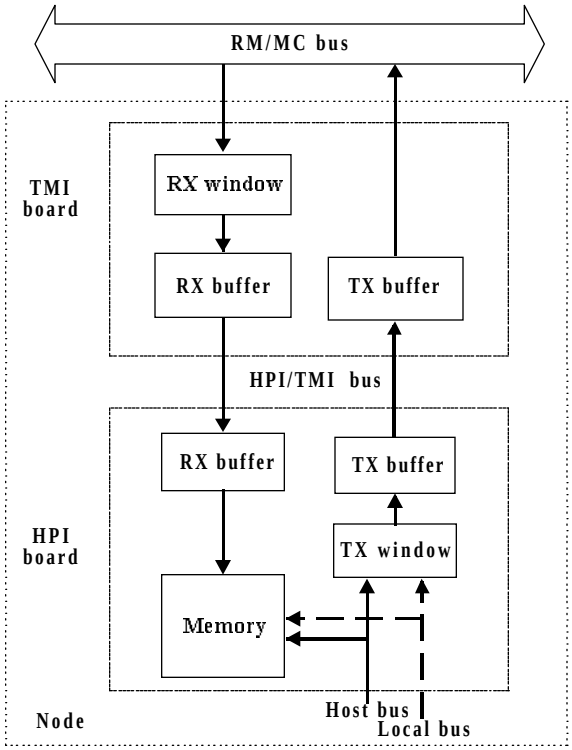


**Figure 1.** An RM/MC node

***Description.*** *Each shared memory write (determined by its 48-bit address which falls into an open transmit window) is written into the local memory, and after the address translation, it is propagated to the RM/MC bus. Other nodes snoop the bus. If the receive window of some particular node is open for particular write, after the address translation, the local memory of that node is updated.*

**Advantages.** The RM/MC system combines potentials for a high bandwidth with low latency time. This system has some advantages because it is bus-based. A broadcast mechanism is easy for implementation. Update messages are small, consisting basically of virtual address and data (there are no special headers and trailers, etc.). Propagation time for update messages is small, because interconnection medium spends only one bus cycle (in the case of RMS where the bus is non-multiplexed) to propagate an update message to all nodes. The system tolerates node failures without service disruption.

**Disadvantages.** The main disadvantage is the low system scalability. Also, the caching of RM is disabled, because the HPI board, as a slave on the system bus, can not initiate update/invalidate

messages upon receiving an update from the RM/MC bus, in order to keep the cache memory consistent. Short (word) messages and long (block) messages share the same FIFO buffers, so the short messages always have to wait after the long ones.

#### *3.1.1. Improved RMS for the PC environment*

In order to use the RMS in the personal computers (PC) market, the University of Belgrade group in cooperation with Encore proposed a solution that better suits higher node counts desirable for a network of PCs [Savi95]. They implemented a prototype board for the EISA bus back in the 1993/94 time frame. The proposed system reduces the RM bus traffic, deals with transient and permanent overloads on the transmit side, as well as on the receive side, and alleviates the problem of memory contention.

#### *3.1.2. RM/MC++ System*

Another project of the University of Belgrade group and Encore proposed a solution that intends to enhance the RM/MC system performance [Jova95], by combining compile- and run-time actions. The system performs better if the probability is minimized that a short message follows with a delay after a long one, in conditions when the two are of the same priority and independent. Update message streams defined at compile time are separated by duplicating resources at critical places. A simple algorithm for moving messages at run-time into the higher priority stream, if allowed by mutual dependencies, was proposed.

#### *3.1.3. Encore LAM*

Strict consistency of RM of the aforementioned systems is not guaranteed by hardware because updates are based on asynchronous, non-instantaneous communication. The consistency problem for projects employing these systems has been solved for each particular system separately. Therefore, Encore (in cooperation with University of Florida) went a step further, once and for all solving this problem by proposing the LAM - a hybrid DSM system established on the RM interconnect hardware (the RM/MC bus), upgraded with a software consistency policy [Dent96].

The LAM uses library routines to manage allocation of the RM space and to synchronize the accesses to this space. The MCM is based on entry consistency. A shared data structure may cross page boundaries. This LAM implementation supports only one memory pool (divided into three regions: data, control, and lock space), although multiple pools could simplify prioritized partitioning of the RM space.

### 3.2. MODCOMP MMM

The MMM (Mirror Memory Multiprocessor) represents an asymmetric, tightly-coupled multiprocessor based on RM concept that delivers both the high computational performance and the real-time performance needed in time-critical applications [Furh97].

**Essence.** The MMM consists of a host computer which controls the operation of the whole system and up to eight tightly coupled nodes housed in a single VME chassis, as shown in Figure 2. Each node consists of a single-board computer and a memory board. Both boards are connected to two buses, the VME and VSB. The memory is divided into private and mirrored segments.

The mirror memory is based on a dual-port memory with logic to create a slave function on the VME bus which supports a broadcast mechanism. The memory contains location monitors which can be programmed to generate interrupts across the VSB bus to the target computer. The combination of these two functions creates data and interrupt broadcast mechanisms.

**Applications.** The MMM supports both hard and soft real-time applications. Examples include, but are not limited to: factory automation, process control, SCADA, data communications, as well as simulation and trainer applications.

**Advantages.** The system incorporates the rate-monotonic scheduling algorithm, which assures that all tasks complete within their deadlines (hard real-time scheduling). The MMM supports a variety of high-performance industry standard I/O buses, interfaces, and protocols.

**Disadvantages.** As other bus based multiprocessors, this system is characterized by poor scalability. Another weakness of this concept is dual function of the VME bus, which serves as a medium for maintaining the RM coherence and also as the system bus.

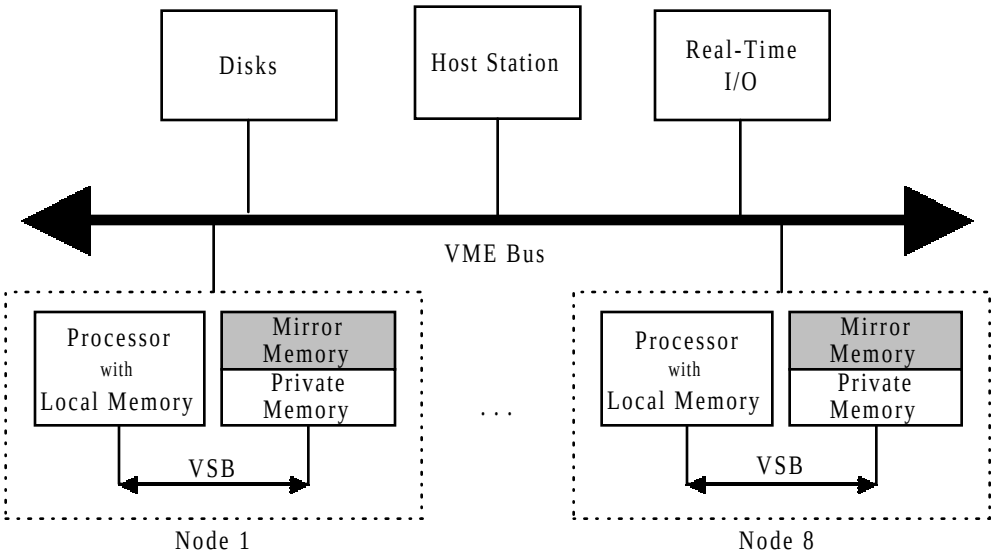


**Figure 2.** Architecture of the MMM

***Description.*** *The VSB bus is used to read from and to write into private memory, as well as to read from the mirrored memory. Writes to the mirrored memory occur across the VME bus.*

## 3.3. Widely Distributed Replicated Shared Memory

A group from the University of Tokyo have proposed an RM model (called widely distributed replicated shared memory) [Oguc95] suitable for the case when a small number of nodes are distributed over a wide area.

**Essence.** Each node (Figure 3) maintains its own copy of the entire contents of global shared memory space on local memory which is on the same board with the processor. Write accesses to this area are broadcast to all other nodes by a multicast server. A prototype of bus-connected RSM has been successfully implemented using multi-threaded programming on multi-CPU SPARC stations.

Updates are not broadcast upon each write. Shared variables accessed in critical sections are broadcast only after a series of exclusive executions has finished. Therefore, the implemented MCM is looser than that of the traditional RM systems, in which the global memory data item is broadcast each time it is written to.

**Applications.** The proposed model is considered to be suitable particularly for applications which impose real time operations in a widely distributed environment, since other latency hiding techniques such as context switching or data prefetching are not effective for real time demands.

**Advantages.** The system supports a kind of release MCM, which is an important step up, in comparison with most other RM approaches. This model has a capability to delay broadcasting of shared memory writes until a series of exclusive executions has completed.

**Disadvantages.** If multicast server is completely realized in software, a software overhead problem will exist. If single processor computer is used as a node, an application execution must be suspended while global data are broadcast. Otherwise, if multiprocessor computer is used as a node, applications can continue their execution on one CPU, while broadcasting is performed by other CPUs.

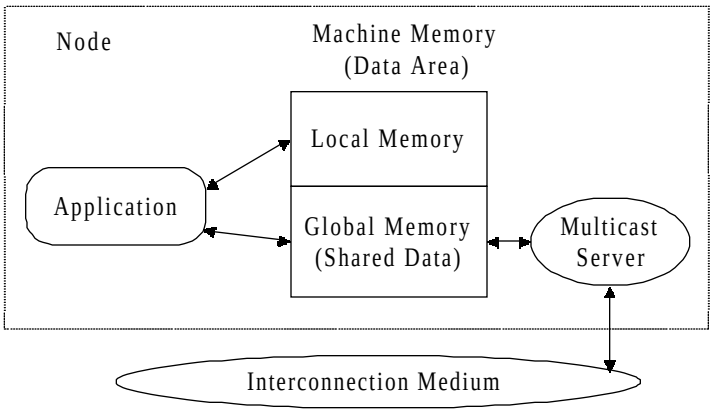


**Figure 3.** A Widely Distributed Replicated Shared Memory node

***Description.*** *Multicast server can be realized in software without any hardware modification of the host computer, while significant performance improvement can be expected by addition of custom hardware.*

## 3.4. Memory Channel Network for PCI

Digital Equipment Corporation has designed a Memory Channel (MC) network for PCI [Gill96] in order to enhance both the parallel performance and high-availability of a cluster. The idea was to map the interconnection network into the virtual address space of individual nodes, so the foundation for this project was the Encore's RM technology.

**Essence.** The MC uses an N-by-N (N=8 or N=16) nonblocking full-duplex crossbar with broadcast capabilities as interconnection network. The data path width is 16 bits, a trade-off that offers a manageable skew and the required high bandwidth. The interconnection medium is either copper or fiber-optic.

This system supports page-level connection granularity. A shared page can be mapped into the process's address space for either reads or writes, but not for both. However, it can be mapped multiple times, either for reads or for writes. Each node (Figure 4) that maps a shared page for reads allocates a page of physical memory. No memory is allocated for write-mapped shared pages, but the

particular page table entry is connected to the appropriate PCI address space. Therefore, a shared write bypasses all caches, while a receiving node can cache a contention of a page mapped for reads.

The basic network primitive provides a write-only connection between a page of virtual address space on the transmitting node and a page of physical memory on the receiving node. Supported connection models are: point-to-point, multicast, and broadcast. Only a clusterwide service can establish connections, ensuring at that time protection and security. Strict ordering of messages is guaranteed under all conditions, even if a transmission error occurs.

**Applications.** The MC is supported by highly optimized PVM (Parallel Virtual Machine), MPI (Message Passing Interface), and HPF (High Performance Fortran) software development tools for parallel applications. Oracle Parallel Server could be used to implement data base applications.

**Advantages.** A server can transmit data directly to the requesting nodes, without affecting its local memory. The caching of RM is supported. Acknowledgments can be requested from receiver nodes. An innovative remote read primitive is implemented as two write transfers without software intervention.

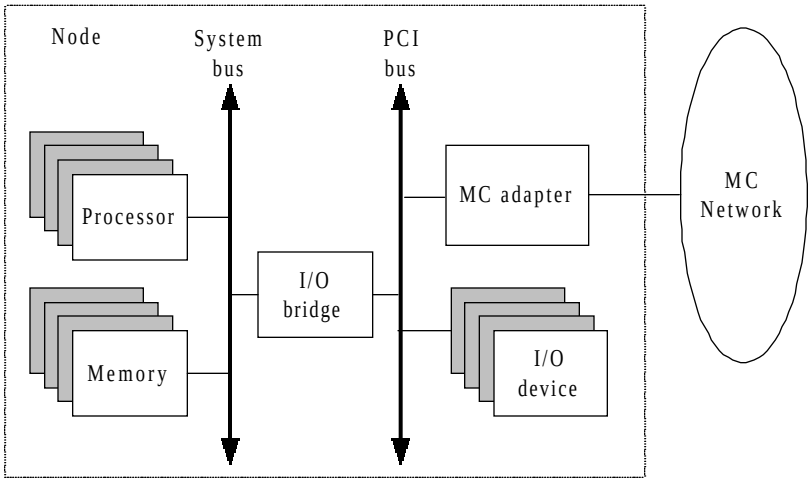


**Figure 4.** Memory Channel network for PCI

***Description.*** *The MC packet consists of the PCI write with a header identifying the destination and trailer containing a 32-bit CRC. Adapter on receiving node, upon receiving such packet, strips off the header and trailer, and sends the write to the PCI bus. Node IO bridge maps the write to the proper physical memory page, just as it would be the case for a normal IO operation.*

**Disadvantages.** This system, designed having homogeneous clusters in mind, does not support heterogeneous computing, in spite of the fact that all computers that incorporate the PCI bus could be connected using this approach. The use of a crossbar limits the number of nodes in the system.

### 3.5. Network Shared Memory

People from Architecture Technology Corporation have proposed a low-cost approach for clustering workstations into single, shared memory mid-range parallel computer [Rama95]. The essence of their proposal, called Network Shared Memory (NSM) system, is RM based on a ring.

**Essence.** Workstations are connected in a unidirectional slotted ring by high speed optical links (Figure 5). The shared memory is implemented within the NMIs (Network Memory Interface) associated with workstations. Each NMI maintains a copy of the entire shared address space. This

memory-to-memory interconnect can be described as a "carousel" of fixed-sized slots that circulate around the ring.

The granularity of memory update is a word. Sequential consistency of RM is guaranteed by hardware since the CPU is blocked from the time a shared memory write is issued until the data has traversed through the ring. The system also provides hardware support for synchronization by implementing a separate synchronization ring.

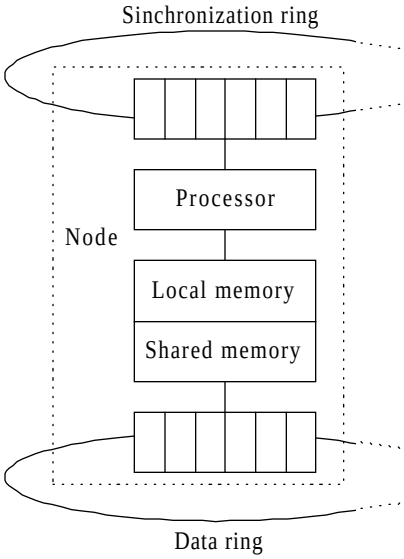


**Figure 5.** A Network Shared Memory node

***Description.*** *When a CPU issues a write operation to shared memory, the associated NMI places it in one of the empty slots in the data ring. As this slot traverses through the ring, each of the other NMIs reads the contents of the slot and updates its copy of the shared memory. After it has fully traversed around the ring, the NMI empties the contents of this slot, and at the end updates its copy of the shared memory.*

**Applications.** This system is intended to be a general purpose mid-range parallel computer, so examples of supported applications are: automotive CAD, pharmaceutical CAD, weather data processing, parallel databases, etc.

**Advantages.** The proposed approach is especially successful for clusters of up to about 60 processors. Results of the performance analysis performed by authors [Rama95] show that it outperforms the targeted competition (directory crossbar based and bus-based symmetric multiprocessors).

**Disadvantages.** In order to ensure memory consistency by hardware, the overlapping computation with communication, a major advantage of RM systems, is not supported. Any relaxed MCM would significantly improve the performance of this system.

### 3.6. SCRAMNet+

The SCRAMNet (Shared Common RAM Network) represents a very popular commercially available RM family of products based on a ring. The Systran Corporation generated the SCRAMNet Classic product line in 1989. It was refined and advanced, evolving into the fully compatible SCRAMNet+ [Syst95]. Because the SCRAMNet+ is the state of the art, only it is described here.

**Essence.** Up to 256 nodes (Figure 6) can be connected via a single fiber optic ring. Each node keeps its own copy of the entire RM space. When deterministic low-latency data transmission is required, the SCRAMNet+ uses small fixed length update packets - an 82-bit message contains a 32-bit useful datum. Besides, variable length packets are provided in order to improve the effective network throughput. These packets have up to 1024 bytes of useful data obtained by merging word writes with subsequent addresses placed in the transmit FIFO.

The SCRAMNet+ is intended to work in a heterogeneous environment. Therefore, hardware byte swapping is supported. Two protocols concerning transmission errors are provided. The first allows error correction only at the receiving nodes, while another supports automatic retransmission of packets. This system employs either a fiber optic cabling (up to 3500 meters), or a coaxial cabling (up to 30 meters) as a transmission medium.

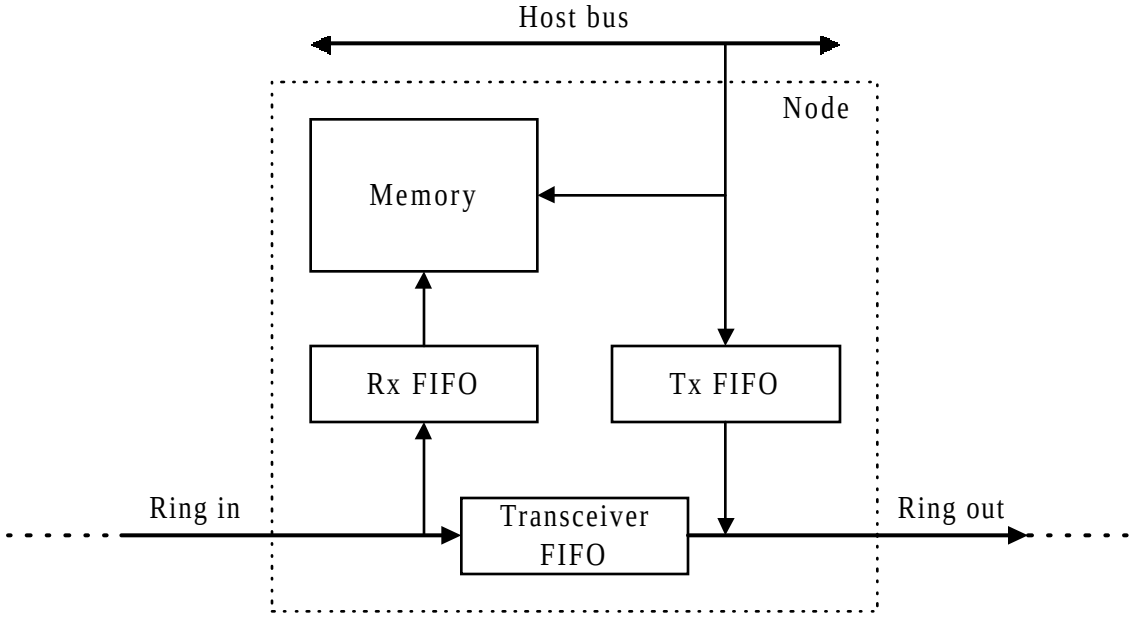


**Figure 6.** A SCRAMNet+ node

***Description.*** *Messages coming into a node from the ring, do not wait for the whole message to be placed into the transceiver FIFO for retransmission, and always have priority over a just generated message (which waits in the transmit FIFO).*

**Applications.** The SCRAMNet+ is optimized for many real-time applications including: simulation (e.g., aircraft, ship, land vehicle, missile), telemetry, robotics, data acquisition, instrumentation and control, and virtual reality.

**Advantages.** The network board can be directly connected to the following system buses: VME, ISA, EISA, Sbus, GIO, PCI, and PCM, while adapters are needed for: MULTIBUS II, SelBUS, and Concurrent EDMA buses. Also, the network board has been designed to be independent from the used medium; the modular media access card, which plugs onto the network board, provides complete media interface. Additional advantage is a kind of data filtering, very useful for cyclical applications (e.g., aircraft flight simulation). Only writes that produce data value changes are broadcast to other nodes.

**Disadvantages.** The RM is limited to 8 MB, due to the fixed number of address lines provided by the standard. Also, there is no capability to dynamically map the RM space and DMA blocks are not supported.

### 3.7. VMIC RM Network

The VMIC (VME Microsystems International Corporation) RM Network [Vmic95] is another commercially available system dedicated to real-time applications. There are two product families distinguished by their transmission medium.

**Essence.** The 5550 family is characterized by 32-bit parallel transfers between nodes over up to 50 feet long flat-ribbon cable (interconnection network is a bus). Up to 16 nodes can be connected. All nodes are given equal priority access to the network. A token-passing scheme determines which node has the access to the cable. Node 0 is the token controller.

The fiber-optic family (5588) very much looks like Systran SCRAMNet+. The communication network is a ring. Up to 16 MB of memory is allowed to be shared among up to 256 independent nodes on distances up to 1000 feet. Data is transmitted over the fiber cable at 1.2 Gbit/sec.

**Applications.** The VMIC RM network is approved for many real-time applications including: aircraft, ship, submarine, and power plant simulators, automated testing systems, aluminum rolling mill control/monitoring, industrial process control, and data acquisition.

**Advantages.** The VMIC RM Network supports a number of popular system buses: VME, ISA, EISA, Sbus, PCI, and PCM. This system is also used to network PLCs (Programmable Logic Controller). Hardware conversion between endian types is supported. A SRAM is used to implement on-board RM, because it provides fast read access time to stored data.

**Disadvantages.** Each node keeps its own copy of the entire RM space. Data filtering is not provided. Interrupts are mostly used for data synchronization across the network, but the use of an interrupt service routine on a receiving node to read the just-received data involves software overhead.

### 3.8. MERLIN and SESAME

The MERLIN (MEmory Routed, Logical Interconnection Network) system represents an open architecture for tightly coupled heterogeneous networks [Witt89]. It was proposed and developed at the Sandia National Laboratories and the SUNY/Stony Brook. The main shortcoming of the Merlin implementation is no hardware support for write-ordering and synchronization. These characteristics, desirable for large network efficiency, are included into its successor SESAME (Scalable Eagerly ShAred MEmory) [Witt92].

**Essence.** The kind of interconnection is not fixed. However, the prototype implements a fiber optic 2-dimensional torodial mesh. The path of an update message is not carried with the data. It is determined by the global virtual address which is not changed while the message is routed through intermediate nodes. The granularity of data sharing is a page. The page size is selectable, but must be the same for all nodes (Figure 7).

The sharing is organized within multicast domains, with centralized control of each domain. Control roots for different domains are spread throughout the network to avoid bottlenecks. Multiple writers within a multicast group are allowed. All writes within one multicast group occur in the same (sequential) order on all participating nodes. Also, sequence numbers are assigned to packets to enforce arrival in the originally written order.

Routes for update messages are determined statically. However, based on a compile analysis, sharing can be dynamically disabled for temporary data changes. The SESAME works basically with single word packets which comprise a 64-bit data item and a 40-bit address. In addition, to gain higher bandwidth, single word packets with consecutive addresses can be merged.

**Applications.** A testing of the prototype has shown that the system performs well on almost all scientific applications. However, simulation results demonstrate that there are applications that scale especially well (parallel Gaussian elimination and Fast Fourier Transform programs).

**Advantages.** The system is intended to scale up efficiently to thousands of processors. Only 10% of the node interface is processor dependent, so this system can easily be ported to numerous different processors. The lock values, which are distributed, can be rapidly set and tested.

**Disadvantages.** Real block transfers generated by DMA are not supported. Since the Gazelle GaAs chips (used to transform words into bit-serial streams for fiber-optic transmission and back) can not correct transmission errors successfully, they must be corrected by sequence numbering and data retransmission.

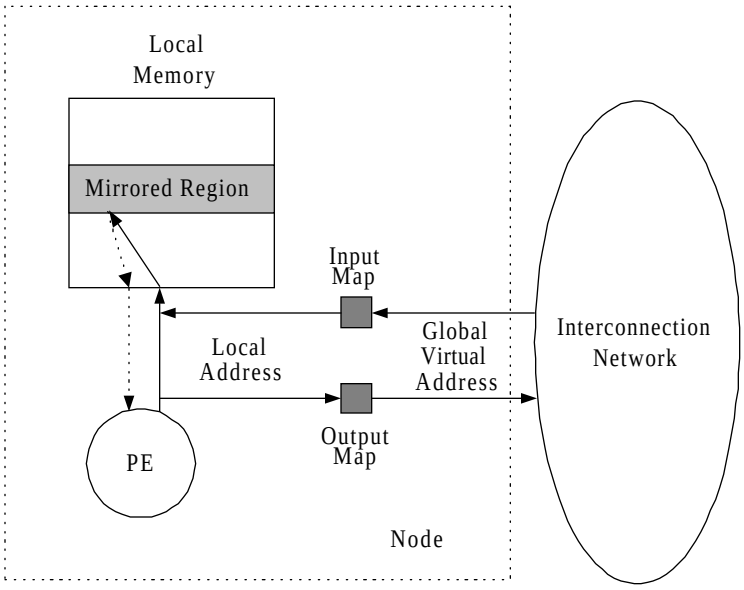


**Figure 7.** A Merlin node

***Description.*** *During each local memory write, the local physical address is checked in the output map table. If the write is shared, the local physical address is translated into the global virtual address which is fixed during the transfer. Each node has the steering map table which uses the global virtual address to route the message to the next node on the way. A receiving node retranslates the global virtual address into the proper local physical address using the input map table and updates the particular memory location.*

## 3.9. SHRIMP

The SHRIMP (Scalable High-performance Really Inexpensive Multi-Processor) project of scalable multicomputers at Princeton University is based on a custom-designed virtual memory mapped network interface [Blum-94]. Actually, two network interface designs were implemented. The first one, providing minimal hardware help, does not support automatic update, and

consequently is out of scope of this paper. The second one, considered here, is intended to provide as much hardware support as needed to minimize communication latency.

**Essence.** The network interface (Figure 8) maps the virtual memory of a sending process to the virtual memory of a receiving process, in such a way that ordinary memory store instructions cause data to be propagated without extra processor involvement. "Sends" to a dominant destination are made cheaper by separating destination specification from data specification in message passing primitives.

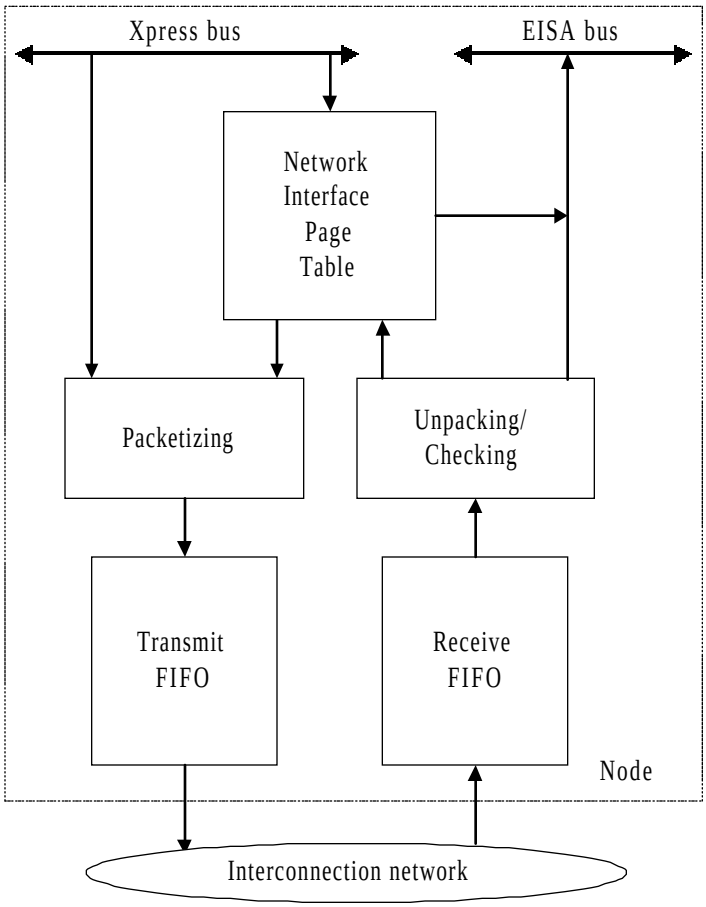


**Figure 8.** The SHRIMP network interface

***Description.*** *The prototype uses Intel Pentium Xpress PC systems as nodes; they are interconnected by an Intel Paragon routing backplane (a two-dimensional mesh). Because the network interface can not master the Xpress bus, the EISA bus is used to update memory from interconnection network. Information needed to maintain the mappings between the physical memories are held in a network interface page table.*

Two transfer strategies are supported: automatic update and deliberate update. There are two kinds of automatic updates: single-write which is optimized for low overhead and block-write which is optimized for efficient network bandwidth usage. Consecutive writes buffered in the transmit FIFO are merged into the same block automatic update packet if they are within the same page and occur within a programmable time limit one from another. Deliberate update, related to blocks only, is propagated only after the sending process issues an explicit send command.

The SHRIMP network interface does not allow mapping of arbitrarily sized segments of memory, but approximates this by allowing a physical page to be split between two separate mappings at some configurable offset. The network interface and the network both maintain FIFO ordering between updates. The SHRIMP project team has proposed two relaxed MCMs: automatic update release consistency and scope consistency.

**Applications.** This system, supporting both word transfers and block transfers, as well as automatic update and deliberate update mechanisms, has great potential to be successfully used in almost all kinds of parallel applications.

**Advantages.** The snooping cache architecture of the Xpress PC system insures that caches remain consistent with main memory. A programming model with many of the properties of shared memory is supported, as well as a simple and flexible message passing mechanism.

**Disadvantages.** The main drawback is that broadcast and multicast communication models are not supported, although certain programs may need to use the same data buffers to communicate with more than one node in the system.

## 3.10. Summary

Differences between the presented RM systems are summarized in Table 2.

| System | Processor Count | Sharing Granularity | RM Mapping | Update Granularity | MCM Protocol | Caching Included | Main Applications |
|---|---|---|---|---|---|---|---|
| Encore RMS | 8 | page | dynamic | word | PC | no | real-time |
| RMS for PC | 16, 32, or more | page | dynamic | word | PC | yes | real-time |
| Encore RM/MC | 8 | page | dynamic | word & block | PC | no | OLTP |
| RM/MC++ | 8 | page | dynamic | word & block | PC | no | OLTP |
| Encore LAM | 8 | segment | dynamic | word & block | EC | no | OLTP |
| Modcomp MMM | 8 | page | dynamic | word | SC | no | real-time |
| RSM | 8 | segment | static | word | RC | no | real-time |
| DEC MC | 16 | page | dynamic | word & block | PC | yes | client-server |
| NSM | 60 | segment | static | word | SC | no | S & E |
| SCRAMNet+ | 256 | segment | static | word | PC | no | real-time |
| VMIC Network | 256 | segment | static | word | PC | no | real-time |
| Sesame | >1000 | page | dynamic | word | PC | no | S & E |
| SHRIMP | >1000 | page | dynamic | word & block | AURC | yes | client-server |

**Table 2.** Differences between represented RM systems

***Description.*** *SC - sequential consistency; PC - processor consistency; RC - release consistency; EC - entry consistency; AURC - automatic update release consistency; S & E - scientific and engineering.*

The strengths and weaknesses of the presented RM systems are summarized in Table 3. This table, also, contains directions for further research of each system.

| System | Advantages | Disadvantages | Possible Improvement Avenues |
|---|---|---|---|
| Encore RMS | simple compared to others | low scalability,<br>high cable complexity,<br>limited number of nodes | caching,<br>data filtering,<br>better control of FIFO overflows |
| RMS for PC | better scalability,<br>data filtering | high cable complexity | more relaxed MCMs,<br>lazy propagation of updates |
| Encore RM/MC | widely applicable | low scalability,<br>high cable complexity,<br>limited number of nodes | caching,<br>hierarchy of RM buses,<br>prioritization |
| RM/MC++ | widely applicable,<br>prioritization,<br>better control of word and block streams | low scalability,<br>compiler has to be modified | caching,<br>hierarchy of RM buses |
| Encore LAM | widely applicable,<br>entry consistency,<br>shared data are organized as segments | one memory pool | caching,<br>multiple memory pools,<br>prioritization,<br>hierarchy of RM buses |
| Modcomp MMM | hard and soft real-time applications | poor scalability,<br>dual function of the VME bus | bus hierarchy,<br>system bus for IO operations |
| RSM | release MCM,<br>lazy updates | RM concept in software,<br>only word updates | dynamic RM mapping,<br>block update granularity |
| DEC MC | extremely high bandwidth,<br>caching,<br>update acknowledgments,<br>remote read primitive,<br>sender could bypass local memory | homogeneous nodes,<br>low number of nodes | heterogeneous computing, relaxed MCM,<br>inclusion of some kind of hierarchy,<br>segment sharing granularity |
| NSM | hardware support for synchronization | no overlapping computation with communication,<br>each node keeps a copy of the entire RM space | relaxed MCM model,<br>dynamic RM mapping |
| SCRAMNet+ | data filtering,<br>modular media interface,<br>heterogeneous computing,<br>merging word updates | RM is limited to 8 MB,<br>each node keeps a copy of the entire RM space | dynamic RM mapping,<br>block update granularity |
| VMIC Network | on-board RM is SRAM,<br>heterogeneous computing,<br>PLCs could be networked | each node keeps a copy of the entire RM space,<br>use of interrupts for data synchronization | dynamic RM mapping,<br>block update granularity,<br>data filtering |
| Sesame | high scalability,<br>heterogeneous computing,<br>merging of word updates,<br>distributed synchronization,<br>dynamic sharing | no real block updates | block update granularity |
| SHRIMP | high scalability, caching,<br>sophisticated relaxed MCMs,<br>deliberate updates, word and block updates, merging of word updates | no broadcast mechanism | hardware broadcast mechanism |

**Table 3.** The strengths and weaknesses of the presented RM systems

***Discussion.*** *Capability of filtering of subsequent writes issued to the same shared address by the same node could reduce the update traffic both on the interconnection network and on the receive side of target nodes. This indirectly could reduce propagation time and memory access time. Lazy propagation of updates would amplify the power of data filtering.*

*Transient overloads on the transmit side of a node could be handled by temporary giving a higher priority to a particular node, while transient overloads on the receive side of a node could be handled by temporary giving a higher priority to the receive FIFO buffer.*

*Hierarchy of RM buses with sophisticated data partitioning schemes and data filtering between stages could improve system scalability. A prioritization scheme could improve system performance by reducing unnecessary waiting, thus making a system even more usable.*

*Dynamic RM mapping enables nodes not to keep a copy of the entire RM space. Therefore, available memory is better utilized. Actually, this mechanism incorporates filtering of updates on the transmit side and the receive side of a node, what could reduce the interconnection network utilization, as well as the memory contention. Block updates also reduce the interconnection network utilization and the memory contention (if the fast page mode is supported).*

*Incorporation of caching of RM regions could significantly reduce the memory access time. A relaxed memory consistency model improves processor utilization. Some hardware support for heterogeneous computing could greatly improve system usability. Segment sharing granularity could improve utilization and partition of memory.*

*A separate system bus for I/O operations leaves the VME bus to handle RM updates only, what could significantly reduce propagation time of update messages.*

*A hardware broadcast mechanism would improve system performance, and could alleviate the implementation of the proposed MCMs.*

# 4. CONCLUSION

The RM/MC system brings together potentials for a high bus bandwidth with low latency time, what makes it useful for many applications. Improved RMS for the PC environment represents a set of relatively inexpensive remedies that cure the scalability inefficiencies of the baseline design. The RM/MC++, combining compile- and run-time actions, intends to enhance the RM/MC system performance. The LAM solves the consistency problem of the aforementioned Encore's systems by involving entry consistency.

The MMM system uses the standard VME bus for maintenance of RM coherence. This system supports hard real-time applications. The RSM, designed for widely distributed environment, considers complete realization of the RM concept in software. This system is capable to delay broadcasting of updates until a series of exclusive executions has completed.

The DEC MC provides an extremely high bandwidth due to a full-duplex crossbar being used as the interconnection network. Strict ordering of messages is provided by hardware. Some kind of hierarchy can increase the low number of nodes in the current implementation. The caching is supported, but heterogeneous computing is not. Remote read primitives are incorporated.

In order to ensure sequential memory consistency completely by hardware, the ring-based NSM does not support the overlapping computation with communication, what is one of main advantages of RM systems. Any relaxed MCM model would significantly improve the performance of this system.

The SCRAMNet+ and VMIC RM Network represent commercially successful real-time network interfaces. These ring-based systems, supporting many state of the art platforms, are particularly

suitable for heterogeneous computing. Also, the SCRAMNet+ provides data filtering and the modular media interface design.

The SESAME and the SHRIMP are academic research projects based on a mesh. Therefore, they are scalable to higher node counts. The SESAME provides strict sequencing of shared writes from multiple writers and the ability to dynamically disable and enable sharing. The SHRIMP supports both word transfers and block transfers, as well as automatic update and deliberate update mechanisms. The caching is provided, but a broadcast mechanism is not. Two relaxed MCMs have been proposed: automatic update release consistency and scope consistency.

Although RM systems commercially exist for a relatively long time, there are a number of topics worth further investigation: more relaxed MCMs suitable for RM systems, lazy propagation of updates, cache coherence mechanisms, trade-offs between software and hardware implementations, and the search for more appropriate applications.

# 5. REFERENCES


[Blum94] M. Blumrich, et al., "Virtual Memory Mapped Network Interface for the SHRIMP Multicomputer," *Proc. 21th Annual International Symposium on Computer Architecture*, IEEE Computer Society Press, Los Alamitos, California, USA, 1994, pp. 142-153.

[Dent96] R. Denton, T. Johnson, "Distributed Shared Memory Using Reflective Memory: The LAM System," *University of Florida Technical Report (TR96-021)*, Gainesville, Florida, USA, July1996.

[Enco90] *RMS Functional Specification*, Encore Computer Corporation, Fort Lauderdale, Florida, USA, 1990.

[Furh97] B. Furht, "Architecture and Performance Evaluation of the MMM," *IEEE TC Computer Architecture Newsletter*, March 1997, pp. 66-75.

[Gert93] I. Gertner, *The Reflective Memory / Memory Channel System Overview*, Encore Computer Systems, Fort Lauderdale, Florida, USA, 1993.

[Gill96] R. Gillett, "Memory Channel Network for PCI," *IEEE MICRO*, February 1996, Vol. 16, No. 1, pp. 12-18.

[Jova95] M. Jovanovic, M. Tomasevic, and V. Milutinovic, "A Simulation-Based Comparison of Two Reflective Memory Approaches," *Proceedings of the 28th Hawaii International Conference on System Sciences* (HICSS-95), IEEE Computer Society Press, Los Alamitos, California, USA, 1995, pp. 140-149.

[Nitz91] B. Nitzberg, V. Lo, "Distributed Shared Memory: A Survey of Issues and Algorithms," *IEEE Computer*, Vol. 24, No. 8, August 1991, pp. 52-60.

[Oguc95] M. Oguchi, H. Aida, and T. Saito, "A Proposal for a DSM Architecture Suitable for a Widely Distributed Environment and its Evaluation," *Proceeding of the Fourth IEEE International Symposium on High Performance Distributed Computing* (HPDC-4), Washington, D.C., USA, August 1995, pp.32-39.

[Prot98] J. Protic, M. Tomasevic, and V. Milutinovic, *Tutorial on Distributed Shared Memory Concepts and Systems*, IEEE Computer Society Press, Los Alamitos, California, USA, 1998.

[Rama95] R. Ramanjuan, J. Bonney, K. Thurber, "Network Shared Memory: A New Approach for Clustering Workstations for Parallel Processing," *Proc. Fourth IEEE International Symposium on High Performance Distributed Computing* (HPDC-4), Washington, D.C., USA, August 1995, pp. 48-56.

[Savi95] S. Savic, et al., "Improved RMS for PC Environments," *Journal of Microprocessors and Microsystems*, Vol. 19, No. 10, December 1995, pp. 609-619.

[Syst95] *SCRAMNet+ Protocol Description*, SYSTRAN Corporation, Dayton, Ohio, USA, 1995.

[Witt89] L. Wittie, C. Maples, "Merlin: Massively Parallel Heterogeneous Computing," *Proceedings of the International Conference on Parallel Processing (ICPP-89)*, St. Charles, Illinois, USA, 1989, pp. 142-150.

[Witt92] L. Wittie, G. Hermannsson, and A. Li, "Eager Sharing for Efficient Massive Parallelism," *Proceedings of the International Conference on Parallel Processing (ICPP-92)*, St. Charles, Illinois, USA, August 1992, pp. 251-255.

[Vmic95] *VMIC's Reflective Memory Network*, VME Microsystems International Corporation, Huntsville, AL, USA, 1995.