

SCI Reflective Memory

(Experimental)

Atle Vesterkjær

Dolphin Interconnect Solutions AS

Olaf Helsets vei 6, N-0621 Oslo, Norway

Phone: (47) 23 16 71 42 Fax: (47) 23 16 71 80

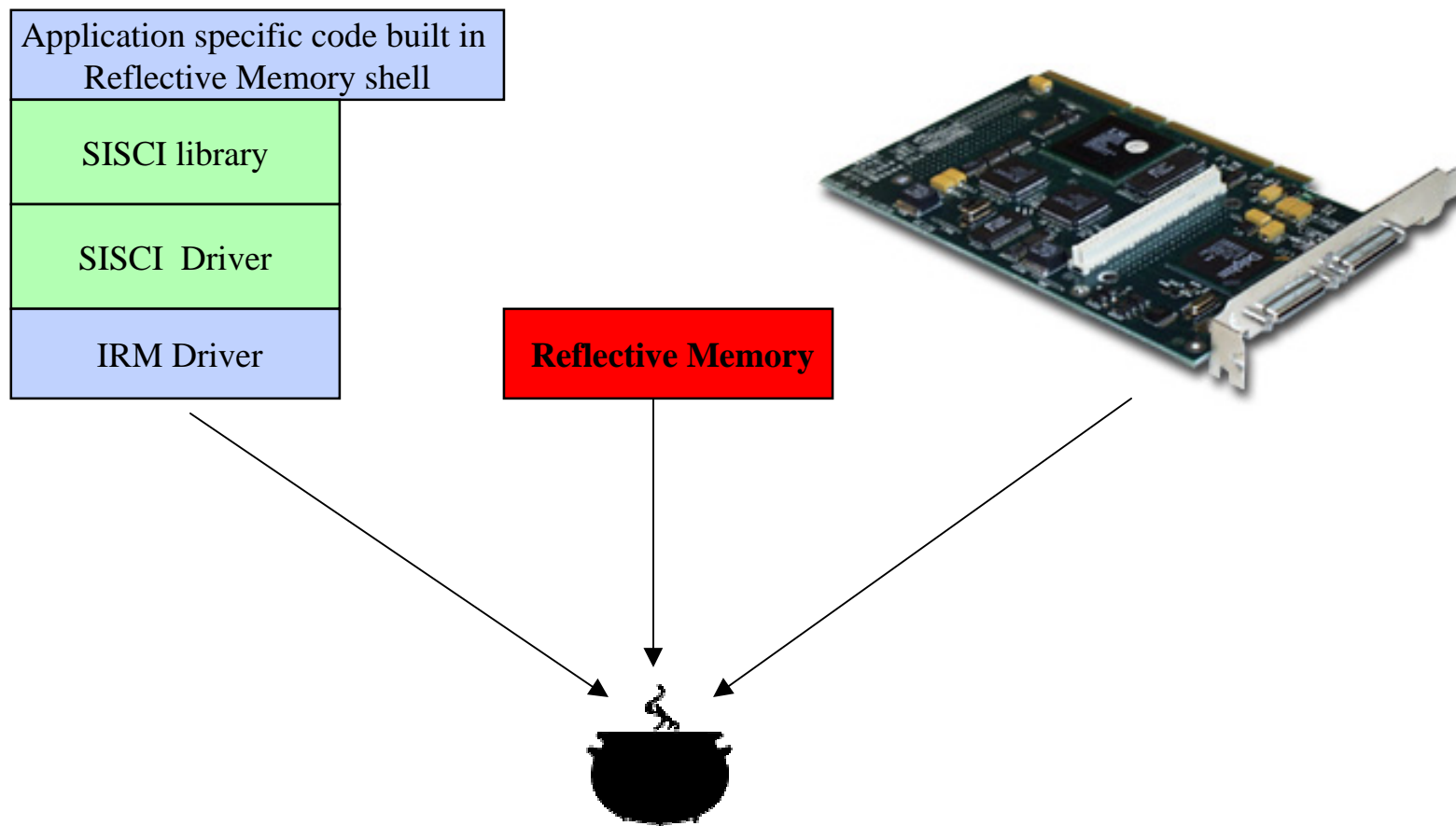
Mail: atleve@dolphinics.no

Introduction



- This presentation aims to give you an idea of how SCI can be used for embedded / realtime solutions.
- SCI Reflective Memory is a software Reflective Memory solution.
- SCI Reflective Memory is a library that you can use to build Reflective Memory applications from, without having to consider the low-level implementation of SCI.

SCI Reflective Memory



Contents



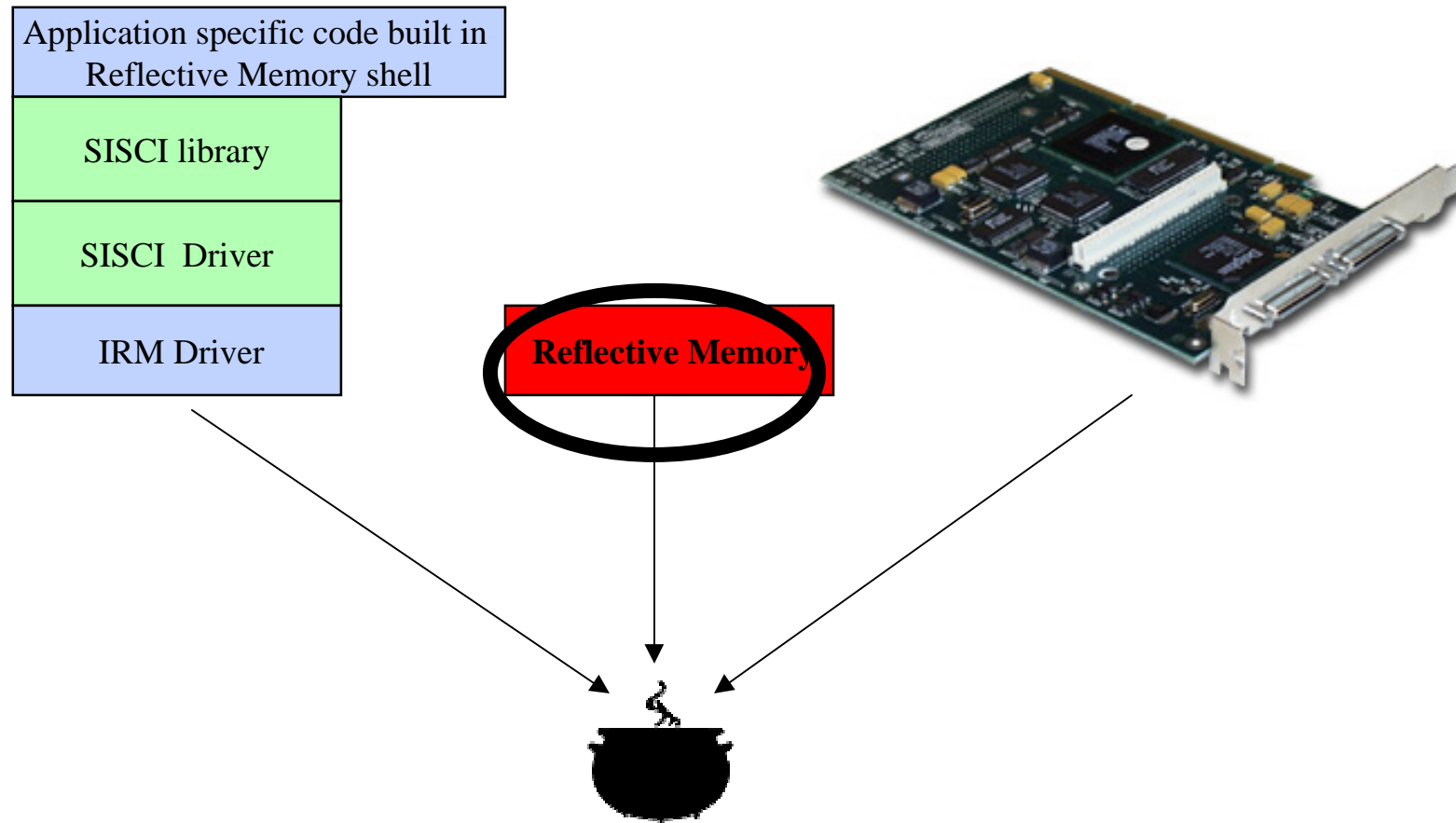
- Introduction to Reflective Memory
- Dolphins HW and SW used in building SCI Reflective Memory
- SCI Reflective Memory technical description, features and benefits

SCI Reflective Memory Lab 1600-1730



- Test and evaluation of SCI Reflective Memory demo programs.
- The exercises are found in your labmanual (one sheet).

Reflective Memory

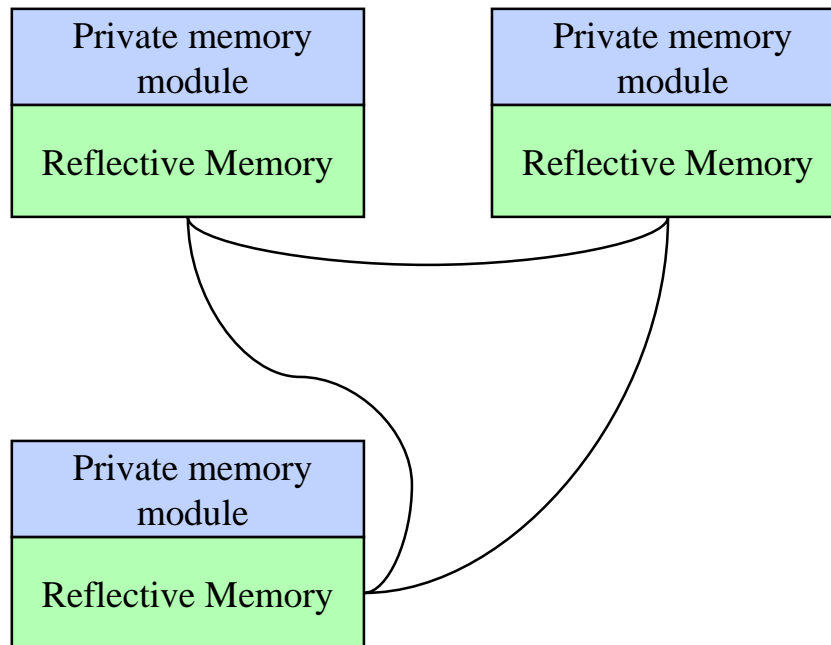


Reflective Memory



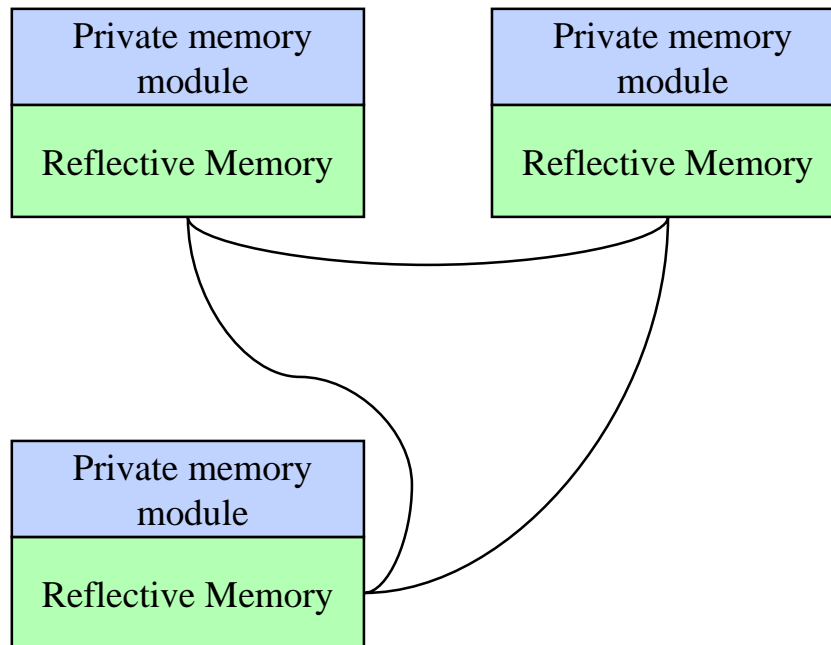
- Reflective Memory systems are a solution to problems raised by message passing in multicomputer environments.
- Reflective Memory systems belong to the class of distributed shared memory systems (DSM)

Reflective Memory



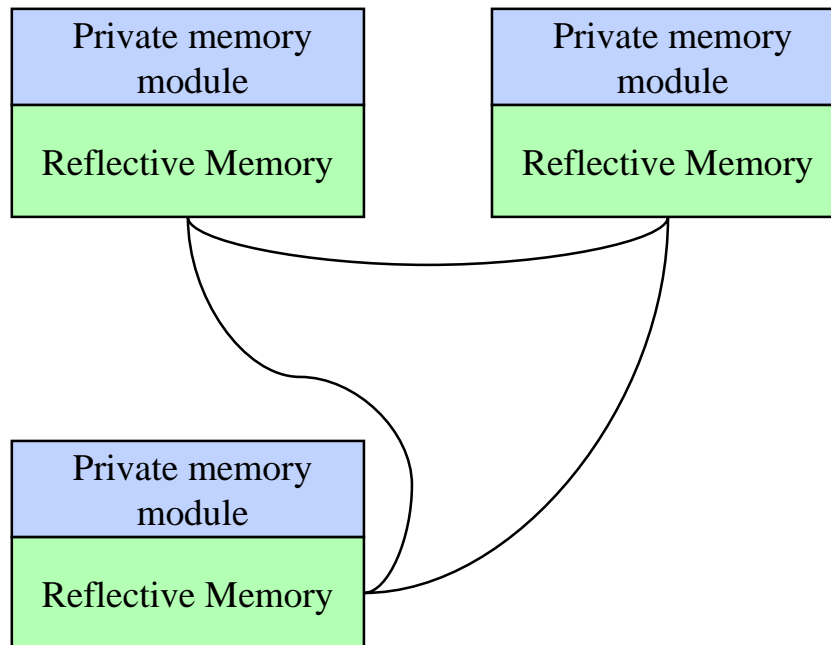
- Each system processor includes a dual-ported local physical memory.
- A part of memory is configured as logically shared.
- The Reflective Memory is composed of all these physically distributed, logically shared memory parts mapped into a global (shared) address space: The Reflective Memory Space.

Reflective Memory



- The main idea of Reflective Memory is that if a shared data item might be reused, an accurate copy of it should be kept in each processors local memory.

Reflective Memory



- Read operations are performed on local memory
- Write operations generates automatic updates of all system copies by a broadcast transaction

Advantages and disadvantages of Reflective Memory systems compared to other DSM systems:



Advantages:

- Computation typically overlaps with communication
- Memory access time is usually constant and thus deterministic.
- Because of their inherent replication they are good for fault tolerance
- Simpler, and have been commercially implemented for decades.
- Read operations are fast.

Disadvantages:

- For applications characterized with longer sequences of writes to the same segments, RM systems may produce unnecessary traffic.
- The interconnection medium usually represent a bottleneck due to many data transfers.
- Processes that write to the same shared memory location must be explicitly synchronized.

Reflective Memory applications

- Aircraft, Ship and Submarine Simulators
- Automated Testing Systems
- Industrial Automation
- High-Speed Data Acquisition

Reflective Memory features



- Reflective Memory updates can occur on any type of interconnect.
- Reflective Memory systems can use any type of topology.
- Reflective Memory systems are not limited by any particular memory consistency model.
- The shared memory regions can be mapped either dynamically or statically.

Typical Reflective Memory features

- ◆ Automatic updates of remote shared memory copies
- ◆ Data filtering: Maybe not every temporarily stored variable have to be reflected?
- ◆ Reflective Memory consistency: The shared region can only be accessed by one party at the time.
- ◆ Only shared writes are propagated through the system

Typical Reflective Memory features

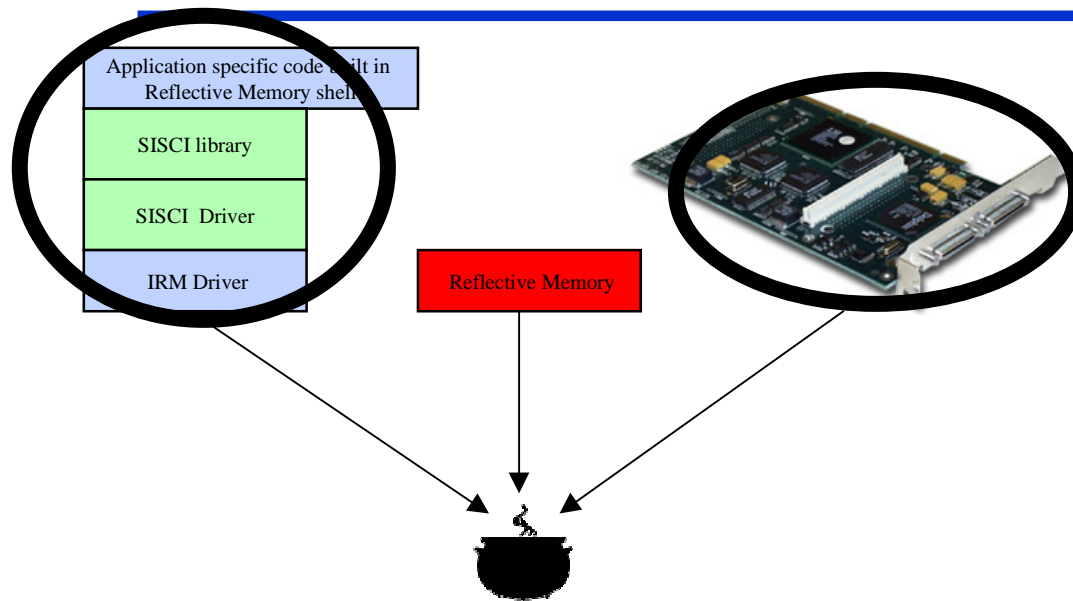
- ◆ one-to-all broadcast communication (hardware based)
- ◆ computation overlaps with communication
- ◆ Hardware support for heterogeneous computing could significantly improve system usability.
- ◆ explicitly synchronization (hardware based): Hardware support for synchronization increase performance.

Why SCI Reflective Memory?



- Reflective Memory is a DSM architecture, like SCI, only organized in another way.
- Reflected Memory could easily be implemented in Dolphin's HW and SW.
- SCI systems have good fault tolerance and redundancy characteristics.
- Competitive performance ratio for Dolphin's SCI products (Will get back to this later).

SCI Reflective Memory



- SCI Reflective Memory is a software reflective memory solution based on Dolphins Adapter cards and software.
- SCI Reflective Memory is a SISCI programming shell that programmers can use to write application specific code for their Reflective Memory application.

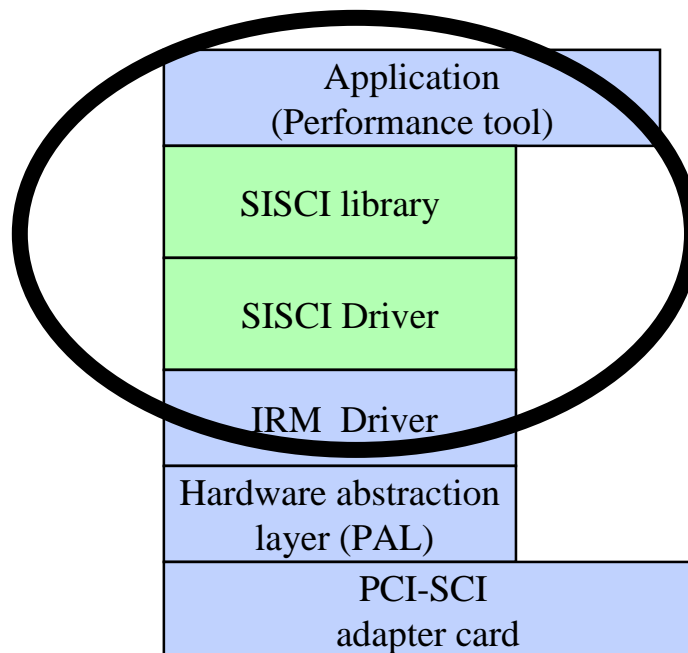
PMC/PCI SCI-64 Adapter Card



- SCI Reflective Memory is a SISI based SCI solution and can be used with all dolphin products that supports SISI.

- Adapter Cards
 - ◆ D307 - SBus
 - ◆ D310 - PCI32
 - ◆ D314 - PMC32
 - ◆ D320 - PCI64
 - ◆ D323 - PMC64
 - ◆ D330 - PCI 66
- Switches
 - ◆ D505 - 4 way (SBus)
 - ◆ D512 - 4 way (PCI)
 - ◆ D515 - 4 - 16 way (PCI)
 - ◆ D525 - 8 way switch

Programming Interface:



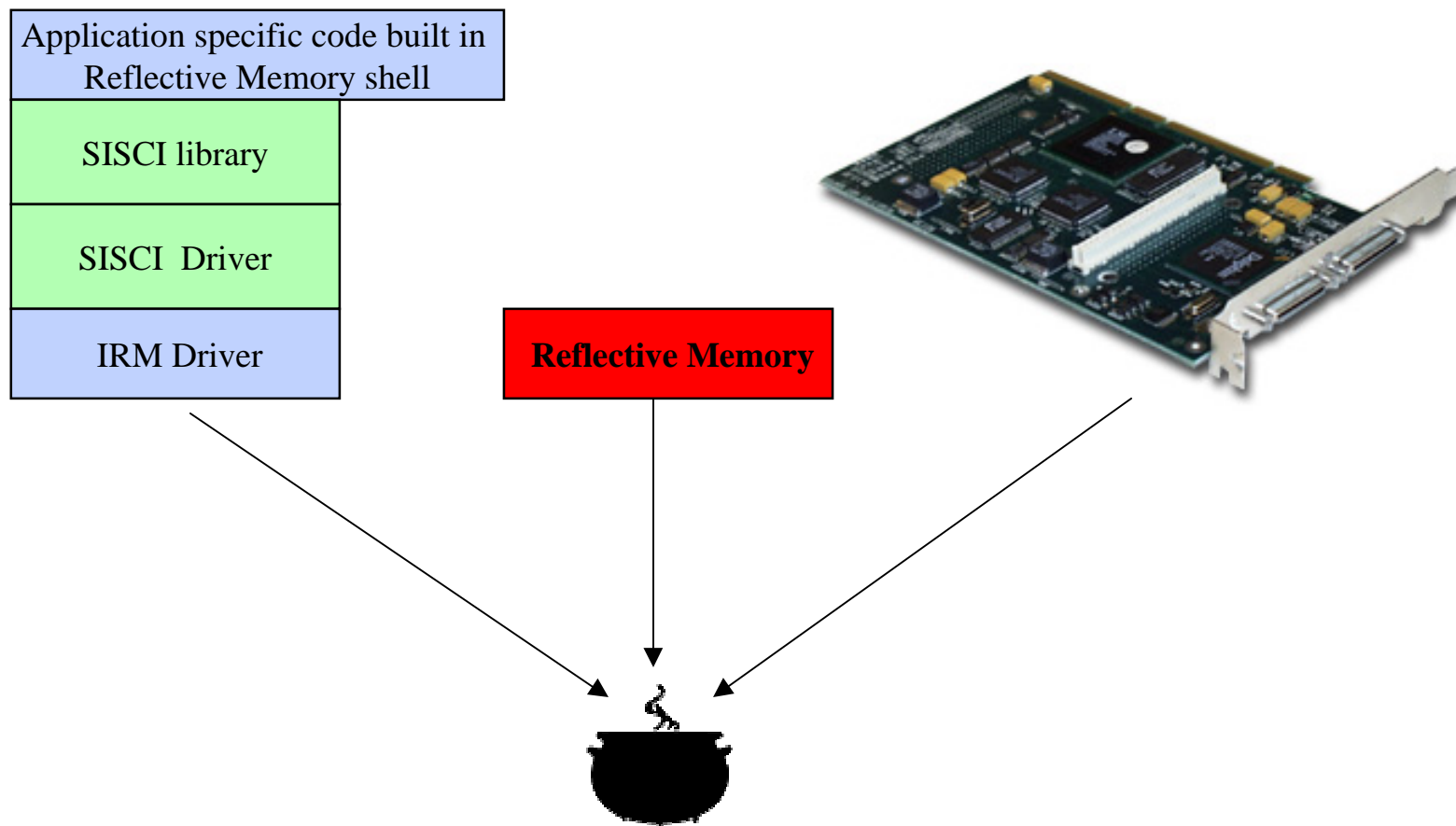
- Application (i.e C-style)
- SISCI API
- SISCI driver
- IRM driver

SISCI features

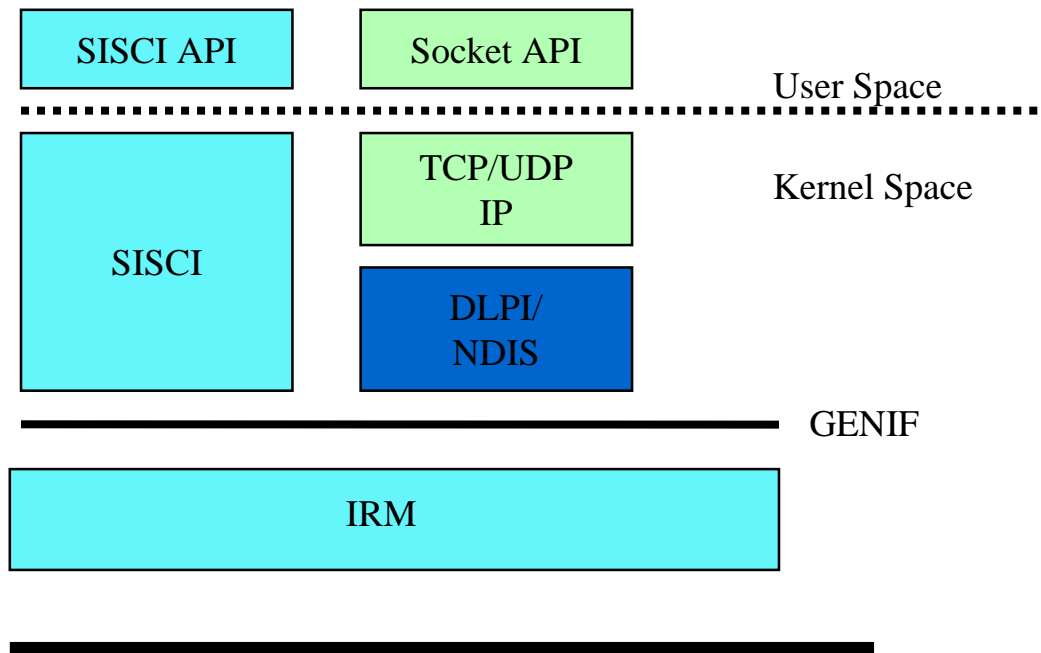


- Access to High Performance HW
- Highly Portable
- Cross Platform / Cross Operating system interoperable
- Simplified SCI Programming
- Flexible
- Reliable Data transfers
- Hostbridge / Adapter Optimization in libraries

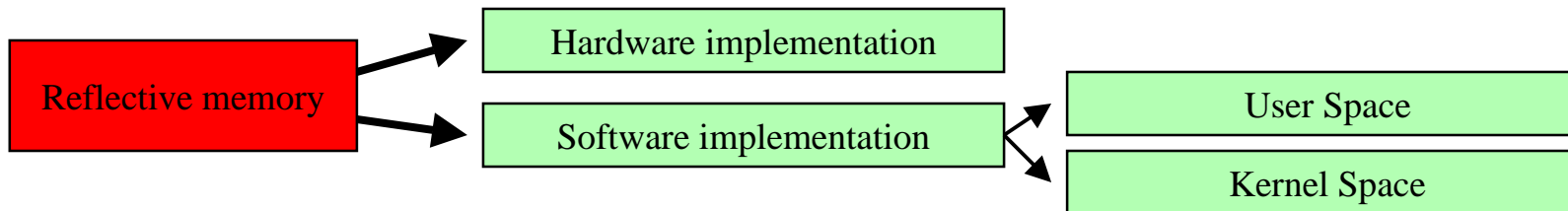
SCI Reflective Memory



SCI Reflective Memory: General



- The first demo of SCI Reflective Memory is implemented for a two node reflective memory configuration.
- The implementation is done in User Space.



SCI Reflective Memory: Overview



- SCI Reflective Memory Library
- SCI Reflective Memory Features
- Reflective Memory Example programs
- Performance

SCI Reflective Memory Library: Overview



- Idea
- Structure
 - ◆ Memory Management
 - ◆ Synchronization
- Applications

SCI Reflective Memory Library: Idea



- A library to build applications from in order to provide a flexible interface to our cards.
- SISCO functions
- Relation to other SISCO C-programs
- Synchronization

SCI Reflective Memory Library: Structure



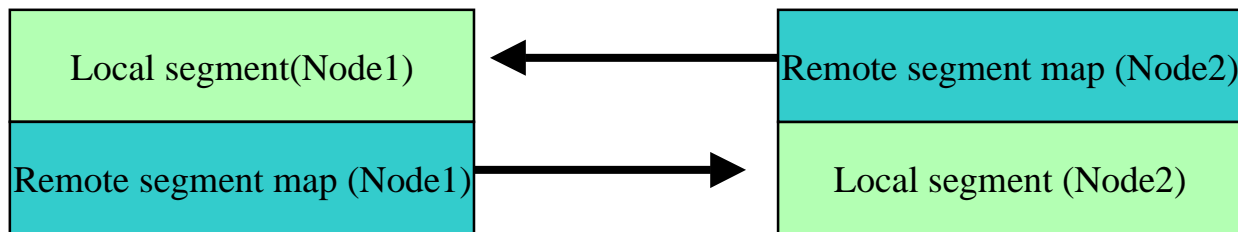
- Memory management
 - ◆ Application specific code should be used for processing, and the SISCi functions for memory access
- Synchronization
 - ◆ In order to guarantee that the local shared reflective memory copies are kept up to date only one node is granted write-access at the time.
 - ◆ Read operations can occur at any time.

SCI Reflective Memory Library: Memory Management



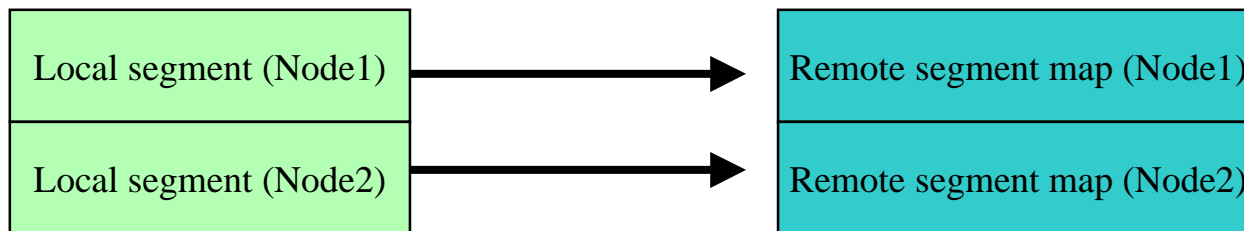
- Segments, duplex mapping.
- Memory read and write operations

SCI Reflective Memory Library: Segments, duplex mapping



- The node preparing to transfer data has to connect to a segment on the node receiving data. In order to get the two nodes to write to each other, they both have to create (at least) one local segment, and they both have to open up a connection to the remote segment (which is created as local on the other node)

SCI Reflective Memory Library: Segments, duplex mapping



- For all RM copies to be uniform, there is a need for an additional mapping as shown above.
- This mapping is carried out by by writing to both the localSegment- and remoteSegment mapping during each write operation. The operations are the same on both nodes.

SCI Reflective Memory Library: Segments, duplex mapping



- **Node1:**

- local-map:

Create, prepare, map
(local), set available

- remote-map:

Connect, map (remote)

- For each write operation to local memory, a write operation to the remote memory is automatically carried out by software.

- **Node2:**

- local-map:

Create, prepare, map
(local), set available

- remote-map:

Connect, map (remote)

- For each write operation to local memory, a write operation to the remote memory is automatically carried out by software.

SCI Reflective Memory Library: Segments, duplex mapping



- If data is written to the Reflective Memory, it is first written into local memory, then transferred to remote memory by any of the SISCi data transfer functions. The programmer is responsible for obeying the strict ordering rule: All write operations to the local memory shall be reflected to the remote memory immediately.

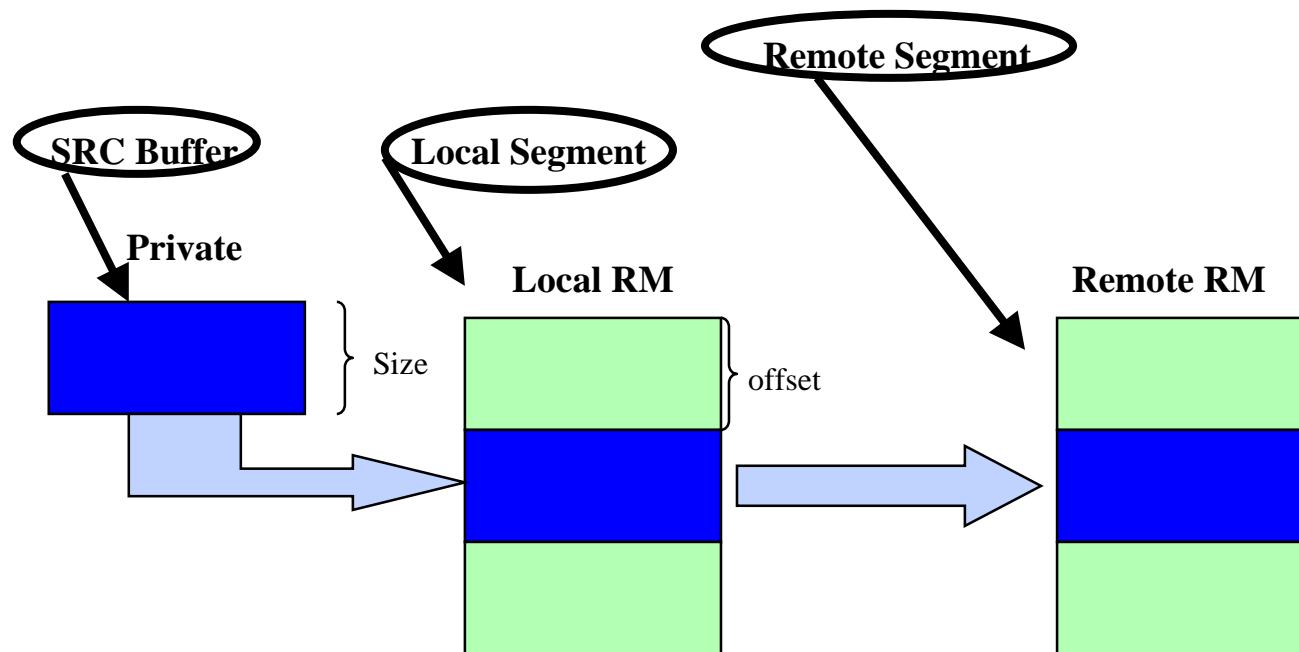
SCI Reflective Memory Library:

Memory read and write operations



- Remote access by SISI functions
 - ★ SCIMemCopy
 - ★ SCITransferBlock
 - ★ SISI DMA Engine
 - ★ *remotePtr = value;
- ◆ Local access by
 - ★ *localPtr=value;
 - ★ memcpy(localBuffer, dummyBuffer, size);

SCI Reflective Memory Library: Data transfer



- A private memory buffer is copied into the Reflective Memory Space
- All three steps are mandatory

SCI Reflective Memory Library: Synchronization



- A central point in a RM system is RM consistency. RM read operations can be performed on local memory, but it should not be possible to have modified data another place in the system. A method that ensures consistent RM copies when nodes are competing for the shared resources is needed. Practically this means that a local access should not be possible when a remote access is in progress, and only one node should have write access to the shared data at the time.

SCI Reflective Memory Library: Synchronization



- Reflective Memory consistency
 - ◆ Polling - asynchronous
 - ◆ Interrupts – timesliced
- Polling is used for better flexibility

SCI Reflective Memory: How to build Reflective Memory applications



- Memory access is taken care of by the reflective memory transfer functions
- Synchronization is used to protect the shared data from corruption

SCI Reflective Memory: Features



- The SCI Reflective Memory is for a two node reflective memory configuration.
- If more nodes shall be supported a modified synchronization scheme has to be implemented. Apart from that there is no other limits in making a multinode SCI Reflective Memory

SCI Reflective Memory: General features



- All nodes share the RM space.
- All nodes have a local copy of the entire RM space.
- The local copies on the subsequent nodes are automatically updated.
- The synchronization logic ensures that only one node has write access to the RM at the time, keeping all RM copies consistent.
- RM write operations are multicasted to all nodes in the system.

SCI Reflective Memory: General features



- computation overlaps with communication: Using DMA transfers for update of remote RM copies enables computation to overlap with communication, when specific flags are set.
- One-to-all multicast communication is used for remote RM updates.
- Shared data regions are organized as segments

SCI Reflective Memory: General features



- Push-only: Only shared write operations are propagated through the system. A write to the local RM is distributed (reflected) to the RM on all nodes. RM read operations are performed on the local RM copy.
- DMA-, block-, memcpy- and shared memory transfers are supported by the SISI API and the SCI Reflective Memory. When building an application the desired transfer mechanism can be selected.

SCI Reflective Memory: Supported OS



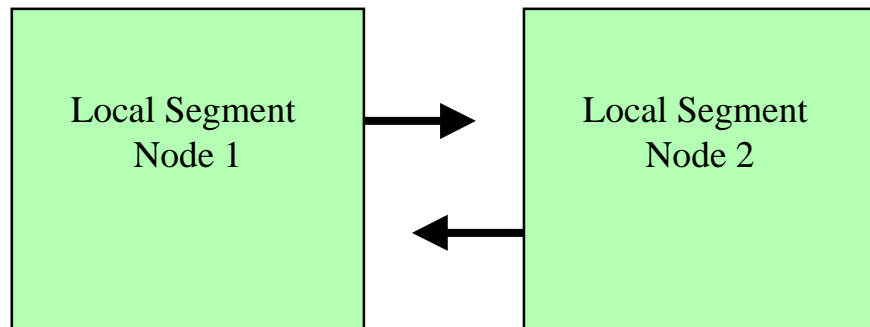
- In general this is just like for the rest of the SISI package, but since SCI Reflective Memory is under development we have not been able to port to all operating systems (OS) yet.
- Currently supported OS are:
 - ◆ Windows (NT & 2000, x86)
 - ◆ Linux (2.2)
 - ◆ Solaris (2.6 / 7, SPARC)
- Next in line of OS that are being ported to:
 - ◆ Lynx
 - ◆ VxWorks (POWERPC)

SCI Reflective Memory example programs



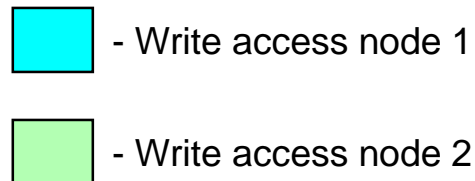
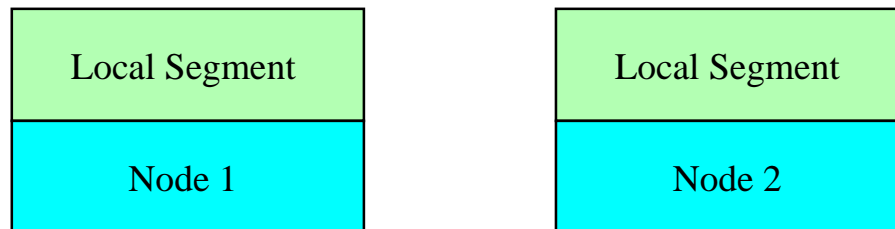
- General Reflective Memory
- Special Reflective Memory
- Multimap Reflective Memory

General Reflective Memory



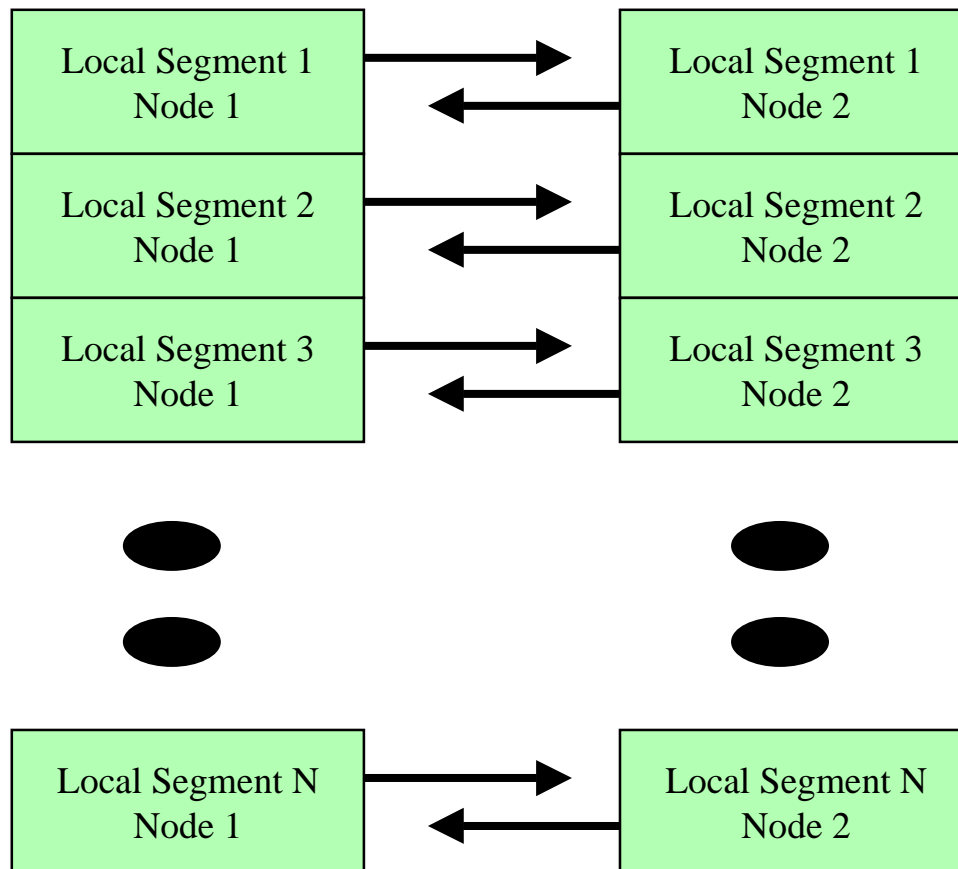
- Only one SISC I segment is created on each node
- The segments are linked together in RM style.

Special Reflective Memory



- Bot nodes have read access to the whole Reflective Memory Space segment, but write access to different halves of the Reflective Memory Space.
- Not really a Reflective Memory solution, but an example of how it can be manipulated for specific applications

Multimap Reflective Memory



- Instead of putting the whole RM space in one segment, the user of `rm_multimap` controls several segments.
- Thus the only time nodes are competing for a resource is when the same segment is requested by more than one (both nodes) at the same time.

How to run the example programs



- In the start-up face of each program you will be asked to enter:
 - ◆ Adapter number
 - ◆ Remote Nodeid
 - ◆ SegmentSize
 - ◆ (Number of segments)
 - ◆ help

How to run the example programs

- These are the available commands:

rm-read: Read from Reflected Memory.

rm-write: Write data to the Reflected Memory.

Special RM write functions:

rm-dma: DMA transfers between two nodes.

rm-block: Block transfers between two nodes.

rm-shmem: Shared memory transfers between two nodes.

rm-memcopy: Transfer data to a previously mapped remote area.

How to run the example programs

- Special RM test functions:

bench-dma: DMA transfers between two nodes. RM style.
bench-block: Block transfers between two nodes. RM style.
bench-shmem: Shared memory transfers between two nodes. RM style.
bench-memcopy: Transfer data to a previously mapped remote area.
bench-full: Test of all RM write-transfers between two nodes.

Special RM test functions where only the remote copy is written to:

single-dma: DMA transfers between two nodes.
single-block: Block transfers between two nodes.
single-shmem: Shared memory transfers between two nodes.
single-memcopy: Transfer data to a previously mapped remote area.
single-full: Test of all RM write-transfers between two nodes.

How to run the example programs

- test-dma: DMA transfers between two nodes, no sync.
- test-block: Block transfers between two nodes, no sync.
- test-shmem: Shared memory transfers between two nodes, no sync.
- test-memcopy: Transfer data to a previously mapped remote area, no sync.
- test-full: Test of all NON-RM write-transfers between two nodes.
- file: Print performance parameters to file
- performance: Print performance parameters for this node
- parameters: Print key parameters for this node
- loops: Number of write-commands in the test routines
- costart: Test with traffic from both nodes starting concurrently
- costop: Disable concurrent start signal
- help: This helpscreen
- q: quit

Performance



- The measurements have been made under the operating system (OS) Windows 2000, but performance is not OS dependent.

SISCI Performance

- Highly dependent of the PC Chipsets
- Latency 2.2 microseconds
- Throughput Application to Application using SISCI
 - ◆ 85 MB/s (33Mhz/32 Bit PCI)
 - ◆ 120 MB/s (33 Mhz/64 Bit PCI)
 - ◆ 240 MB/s* (66 Mhz/64 Bit PCI)

Performance



- The characteristics of the test machines were:
 - ◆ DELL PowerEdge 6300
 - ◆ Pentium II Xeon
 - ◆ CPU clock 400 MHz
 - ◆ 256 MB RAM
 - ◆ 512 KB Level 2 Cache Memory
 - ◆ 440 NX PCI Chipset
 - ◆ Four system processors

Performance (one-way)



- The throughput of remote write operations
- The throughput of a loop containing RM synchronization and remote write operations.
- The throughput of a loop containing RM synchronization, local write operations and remote write operations. RM-style

Performance (one-way)



RM SCIMemCopy transfers without writing to the local segment:

Segment size:	Latency:	Throughput:
524288	5331.96 us	93.77 MB/s
262144	2645.78 us	94.49 MB/s
131072	1329.71 us	94.01 MB/s
65536	672.17 us	92.98 MB/s
32768	343.92 us	90.86 MB/s
16384	179.37 us	87.11 MB/s
8192	97.49 us	80.13 MB/s
4096	56.49 us	69.15 MB/s
2048	36.04 us	54.20 MB/s
1024	25.76 us	37.92 MB/s
512	20.57 us	23.73 MB/s
256	17.95 us	13.60 MB/s
128	16.30 us	7.49 MB/s
64	13.92 us	4.38 MB/s

Performance



RM SCIMemCopy transfers:

Segment size:	Latency:	Throughput:
524288	9953.69 us	50.23 MB/s
262144	3436.81 us	72.74 MB/s
131072	1704.31 us	73.34 MB/s
65536	853.20 us	73.25 MB/s
32768	428.55 us	72.92 MB/s
16384	221.69 us	70.48 MB/s
8192	105.26 us	74.22 MB/s
4096	58.48 us	66.80 MB/s
2048	37.67 us	51.85 MB/s
1024	26.29 us	37.15 MB/s
512	21.23 us	23.00 MB/s
256	18.53 us	13.18 MB/s
128	16.49 us	7.40 MB/s
64	14.02 us	4.35 MB/s

Performance



NON-RM SCIMemCopy transfers:

Segment size:	Latency:	Throughput:
524288	5337.58 us	93.68 MB/s
262144	2639.14 us	94.73 MB/s
131072	1320.58 us	94.66 MB/s
65536	663.29 us	94.23 MB/s
32768	334.80 us	93.34 MB/s
16384	170.47 us	91.66 MB/s
8192	88.66 us	88.12 MB/s
4096	47.61 us	82.05 MB/s
2048	27.18 us	71.87 MB/s
1024	16.87 us	57.89 MB/s
512	11.77 us	41.49 MB/s
256	9.16 us	26.66 MB/s
128	7.84 us	15.57 MB/s
64	4.97 us	12.29 MB/s

Performance (Transfer in both directions simultaneously)



- The throughput of remote write operations
- The throughput of a loop containing RM synchronization and remote write operations.
- The throughput of a loop containing RM synchronization, local write operations and remote write operations. RM-style

Performance



RM SCIMemCopy transfers without writing to the local segment:

Segment size:	Latency:	Throughput:
524288	8374.53 us	119.40 MB/s
262144	4190.23 us	119.33 MB/s
131072	2095.92 us	119.32 MB/s
65536	1053.26 us	118.74 MB/s
32768	528.37 us	118.31 MB/s
16384	269.90 us	115.79 MB/s
8192	139.10 us	112.35 MB/s
4096	74.64 us	104.75 MB/s
2048	42.96 us	91.12 MB/s
1024	27.95 us	70.01 MB/s
512	21.41 us	45.64 MB/s
256	18.41 us	26.54 MB/s
128	16.77 us	14.59 MB/s
64	14.08 us	8.69 MB/s

Performance



RM SCIMemCopy transfers:

Segment size:	Latency:	Throughput:
524288	10945.86 us	91.35 MB/s
262144	4692.67 us	106.48 MB/s
131072	2412.39 us	103.72 MB/s
65536	1154.77 us	108.26 MB/s
32768	606.89 us	103.02 MB/s
16384	312.37 us	100.05 MB/s
8192	146.39 us	106.77 MB/s
4096	77.05 us	101.41 MB/s
2048	44.27 us	88.27 MB/s
1024	28.84 us	67.88 MB/s
512	22.07 us	44.27 MB/s
256	19.02 us	25.71 MB/s
128	17.01 us	14.36 MB/s
64	14.17 us	8.63 MB/s

Performance



NON-RM SCIMemCopy transfers:

Segment size:	Latency:	Throughput:
524288	8369.08 us	119.48 MB/s
262144	4183.97 us	119.51 MB/s
131072	2089.99 us	119.62 MB/s
65536	1043.58 us	119.80 MB/s
32768	519.83 us	120.25 MB/s
16384	260.53 us	120.01 MB/s
8192	130.05 us	120.16 MB/s
4096	65.20 us	119.86 MB/s
2048	33.23 us	117.79 MB/s
1024	18.69 us	104.66 MB/s
512	12.13 us	80.75 MB/s
256	9.40 us	52.07 MB/s
128	7.91 us	30.96 MB/s
64	5.05 us	24.36 MB/s

Future Plans



- We are working in finding partners that are interested in joining us in developing an application based on SCI Reflective Memory for them.
- PSB66 release
- Dig deeper into kernel space and/or hardware to optimize performance and ease of use

Key statement



- The industry leading throughput, and latency of Dolphins interconnect solutions will soon be available for the Reflective Memory market.

Important terms



- We hope that you now will understand the meaning of the terms:
 - ◆ Reflective Memory
 - ◆ PMC/PCI Adapter Cards
 - ◆ SISI
 - ◆ SCI Reflective Memory transfer functions
 - ◆ SCI Reflective Memory synchronization
 - ◆ SCI Reflective Memory duplex mapping of segments

Questions?



Thank you for listening to this presentation! See you in the Lab in half an hour!



SCI Reflective Memory

Atle Vesterkjær

Dolphin Interconnect Solutions AS

Olaf Helsets vei 6, N-0621 Oslo, Norway

Phone: (47) 23 16 71 42 Fax: (47) 23 16 71 80

Mail: atleve@dolphinics.no

Dolphin Interconnect Solutions AS