

Table of Contents

Part I. SCI and Competitive Interconnects for Cluster Computing

1. The SCI Standard and Applications of SCI

Hermann Hellwagner	3
1.1 Introduction	3
1.2 SCI Overview	4
1.2.1 Background	4
1.2.2 Goals	4
1.2.3 Concepts	6
1.2.4 Discussion	11
1.3 The SCI Standard and Some Extensions	11
1.3.1 Logical Layer	12
1.3.2 Cache Coherence Layer	19
1.3.3 Extensions	22
1.4 Applications of SCI	23
1.4.1 System Area Network for Clusters	23
1.4.2 Memory Interconnect for Cache-Coherent Multiprocessors	26
1.4.3 I/O Subsystem Interconnect	30
1.4.4 Large-Scale Data Acquisition System	31
1.5 Related Communication Networks and Concepts	31
1.6 Concluding Remarks	34

2. A Comparison of Three Gigabit Technologies: SCI, Myrinet and SGI/Cray T3D

Christian Kurmann, Thomas Stricker	39
2.1 Introduction	39
2.2 Levels of Comparison	40
2.2.1 Direct Deposit	41
2.2.2 Message Passing (MPI/PVM)	42
2.2.3 Protocol Emulation (TCP/IP)	44
2.3 Gigabit Network Technologies	45
2.3.1 The Intel 80686 Hardware Platform	46
2.3.2 Myricom Myrinet Technology	47

2.3.3	Dolphin PCI-SCI Technology	48
2.3.4	The SGI/Cray T3D – A Reference Point	48
2.3.5	ATM: QoS – But Still Short of a Gigabit/s	50
2.3.6	Gigabit Ethernet – An Outlook	50
2.4	Transfer Modes	51
2.4.1	Overview	51
2.4.2	“Native” and “Alternate” Transfer Modes in the Three Architectures	54
2.5	Performance Evaluation	56
2.5.1	Performance of Local Memory Copy	58
2.5.2	Performance of Direct Transfers to Remote Memory ..	58
2.5.3	Performance of MPI/PVM Transfers	61
2.5.4	Performance of TCP/IP Transfers	64
2.5.5	Discussion and Comparison	65
2.6	Summary	67

Part II. SCI Hardware

3. Dolphin SCI Adapter Cards

Marius Christian Liaaen, Hugo Kohmann	71
3.1 Introduction	71
3.2 Overview of the Adapter Cards	71
3.3 Operating Modes of the SCI Cards	73
3.4 SCI Requester	74
3.4.1 Address Mapping	74
3.4.2 SCI Transaction Handling	75
3.4.3 SCI Packet Requester	77
3.5 SCI Responder	78
3.5.1 Mailbox	79
3.5.2 Access Protection	79
3.5.3 Atomic Access	79
3.5.4 Host Bridge Capabilities	80
3.6 DMA Transfers	80
3.6.1 DMA Transfers on the SBus Card	80
3.6.2 DMA Transfers on the PCI Card	80
3.7 Interrupter	81
3.8 Concurrency Issues	81
3.8.1 Write Assembly	81
3.8.2 Efficient Store Barrier	81
3.9 Performance	82
3.10 Applications and Topologies	82
3.10.1 SAN Interface Adapter	83
3.10.2 Remote I/O Connection and Data Acquisition	83

3.10.3	Switches and Topologies	83
3.11	Cluster Software	85
4.	The TUM PCI/SCI Adapter	
	Georg Acher, Wolfgang Karl, Markus Leberecht	89
4.1	Introduction	89
4.2	The PCI/SCI Adapter Architecture	90
4.3	SCI Packet Encoding and Decoding	92
4.3.1	Overview of Packet Processing	92
4.3.2	Choosing the Technology	92
4.3.3	Internal Structure of the FPGA	93
4.3.4	Structure of the Packet Manager as a Microcode Sequencer	95
4.3.5	Microcode Examples	97
4.3.6	Benefits of the Micro Sequencer	98
4.4	The SCI Unit	99
4.5	Preliminary Results for the PCI/SCI Adapter	99
4.6	Related Work	100
4.7	Conclusion	100

Part III. Interconnection Networks with SCI

5. Low-Level SCI Protocols and Their Application to Flexible Switches

Andreas C. Döring, Wolfgang Obelöer, Gunther Lustig, Erik Maehle 105

5.1	Introduction	105
5.2	Data Format of SCI Packets	105
5.3	Flow Control	107
5.3.1	Flow Control in Rings	107
5.3.2	Packet Sequence in SCI	108
5.3.3	Determination of State Transitions	109
5.4	Bandwidth Multiplexing	110
5.4.1	Bandwidth Management in One Ring	110
5.4.2	Idle Symbols	112
5.4.3	Time-Out Determination	113
5.5	Network Interface	113
5.5.1	Requirements	114
5.5.2	Products	114
5.6	Routers	115
5.6.1	Requirements	115
5.6.2	Products and Challenges	116
5.6.3	Flexible Router	117
5.6.4	Strip-off Decision	118

5.6.5	Routing Decision and Topology	119
5.7	Rule-Based Routing	120
5.8	Conclusion and Outlook	121
6.	SCI Rings, Switches, and Networks for Data Acquisition Systems	
	Harald Richter, Richard Kleber, Matthias Ohlenroth	125
6.1	Introduction	125
6.2	SCI-based Data Acquisition Systems	126
6.3	SCINET Test Beds	127
6.4	Measurement Results	129
6.5	SCI Switches	134
6.6	Efficient Use of SCI Switches	136
6.7	Multistage SCI Networks	139
6.8	Simulation Results	141
6.9	Summary and Conclusions	146
7.	Scalability of SCI Ringlets	
	Geir Horn	151
7.1	Do SCI Ringlets Scale in Number of Nodes?	151
7.2	Ringlet Bandwidth Model	152
7.2.1	Transaction Formats	152
7.2.2	Packet Generation	155
7.2.3	Address Distribution	155
7.2.4	Locality	156
7.2.5	Bypass Rate	157
7.2.6	Echo Packet Rate	158
7.2.7	Output Link Utilization Factor	160
7.3	Scalability Evaluation	160
7.3.1	Common Assumptions	161
7.3.2	Uniform Ringlet Traffic	162
7.3.3	Non-uniform Ringlet Traffic	162
7.3.4	Changing Packet Lengths	163
7.4	Discussion	163
7.5	Conclusion	165
8.	Affordable Scalability Using Multi-Cubes	
	Håkon Bugge, Knut Omang	167
8.1	Introduction	167
8.2	Interconnect Overview	168
8.3	Methodology	168
8.4	Analysis	170
8.4.1	“Hot-Link” Analysis	170

8.4.2	“Hot-B-Link” Analysis	171
8.5	Results	172
8.6	Conclusions	174

Part IV. Device Driver Software and Low-Level APIs

9. Interfacing SCI Device Drivers to Linux

	Roger Butenuth, Hans-Ulrich Heiss	179
9.1	Introduction	179
9.2	Layers of Functionality	180
9.2.1	Address Spaces	180
9.2.2	Levels of Hardware Abstraction	180
9.2.3	Resource Management	182
9.2.4	Virtual Mapping	183
9.2.5	Robustness	184
9.3	Why Linux?	185
9.4	Interfaces of the Driver	186
9.4.1	Hardware	186
9.4.2	Linux	187
9.4.3	User Processes	188
9.4.4	SCI Drivers on Other Nodes	188
9.5	Conclusions	189

10. SCI Physical Layer API

	Volker Lindenstruth, David B. Gustavson	191
10.1	Introduction	191
10.1.1	Scope of the Standard	192
10.2	SCI Physical Layer API Architecture and Features	193
10.2.1	Exception Handling	195
10.2.2	Endianness	195
10.3	Supported Data Types	196
10.4	Miscellaneous Procedures	196
10.5	Address Translation Model	197
10.5.1	Global Object Identifier	199
10.5.2	SCI Global Address Resolution	200
10.6	Shared Memory Transactions	200
10.7	Packet Transactions	202
10.8	Block Transactions	202
10.9	Message Passing Transactions	203
10.10	Cache Transactions	204
10.11	Conclusions	205

Part V. Message Passing Libraries

11. SCI Sockets Library

Hermann Hellwagner, Josef Weidendorfer	209
11.1 Introduction	209
11.1.1 Rationale	209
11.1.2 Overview	210
11.2 Features and Design	210
11.2.1 Features	210
11.2.2 Components	211
11.2.3 Communication via the SSLib	212
11.2.4 Connection Setup	214
11.2.5 Handling Special System Calls	216
11.2.6 Other Calls Intercepted and Handled by the SSLib ...	218
11.2.7 Out-of-Band Data	218
11.3 Implementation Aspects	218
11.3.1 Communication Among Components	218
11.3.2 SSLib Layers	219
11.3.3 Choice of Most Efficient Communication Mechanism ..	220
11.3.4 SSLib Implementations	221
11.3.5 Control Transfers	221
11.4 Functional Tests and Performance	222
11.5 Related Work	224
11.6 Conclusions	227

12. TCP/IP over SCI under Linux

Hüseyin Taskin, Roger Butenuth	231
12.1 Introduction	231
12.2 SCIP Structure	232
12.2.1 Packet Driver Interface	232
12.2.2 Hardware Address Resolution	232
12.2.3 Other Implementation Issues	233
12.3 Performance	234
12.3.1 Configuration	234
12.3.2 Latency	234
12.3.3 Throughput	235
12.4 Conclusion	237

13. PVM for SCI Clusters

Markus Fischer, Alexander Reinefeld	239
13.1 Overview	239
13.2 Parallel Virtual Machine	239

13.2.1	PVM Implementations	240
13.2.2	Models for Zero-Memory-Copy Data Transfer	241
13.3	SCI Communication Model	242
13.4	PVM-SCI	243
13.4.1	System Architecture	243
13.4.2	Supporting Multiple Interconnects	245
13.4.3	Reducing Memory Copies	245
13.4.4	Ring Buffer Management	246
13.4.5	Performance Results	247
13.5	Conclusions	247
14.	ScaMPI – Design and Implementation	
	L.P. Huse, K. Omang, H. Bugge, H. Ry, A.T. Haugsdal, E. Rustad	249
14.1	Introduction	249
14.2	Scali Systems	249
14.3	The SCI Memory Model	250
14.3.1	Coordinating Use of Shared Locations	251
14.3.2	Ensuring Safe Data Transport in SCI – Checkpointing	252
14.3.3	Shared Address Space Programming without the Drawbacks	252
14.4	ScaMPI Design Goals	253
14.5	ScaMPI Implementation	254
14.5.1	Fault Tolerance	254
14.5.2	User Friendliness	256
14.5.3	Third Party Software	256
14.6	Performance Results	257
14.6.1	Barrier	258
14.6.2	All-to-All Communication	259
14.7	Conclusions	260

Part VI. Shared Memory Programming Models and Runtime Mechanisms

15.	Shared Memory vs Message Passing on SCI: A Case Study Using Split-C	
	Max Ibel, Michael Schmitt, Klaus Schausser, Anurag Acharya	267
15.1	Introduction	267
15.1.1	Introduction to Split-C	268
15.1.2	Introduction to Active Messages	269
15.2	Message-Passing Implementation	269
15.2.1	Active Messages on Top of SCI	269
15.2.2	Split-C on Top of Active Messages	272
15.3	Shared Memory Implementation	273

15.3.1	Split-C on Top of SCI	273
15.4	Experimental Evaluation	274
15.4.1	Micro-benchmarks	274
15.4.2	Application Benchmarks	276
15.5	Hybrid Implementation	277
15.5.1	Basic Framework	277
15.5.2	Mapping Strategies	278
15.6	Conclusions	279
16.	A Shared Memory Programming Interface for SCI Clusters	
	Marcus Dormanns, Karsten Scholtysik, Thomas Bemmerl	281
16.1	Introduction	281
16.2	Platform Properties: System Image and Memory Model	282
16.2.1	System Image and Operational Model	282
16.2.2	Memory Model	283
16.3	User Front-End	284
16.4	The Application Programmer's Interface	284
16.4.1	Initialization and Execution Environment	286
16.4.2	Memory Management	286
16.4.3	Synchronization	288
16.4.4	Loop Scheduling	288
16.5	Conclusions	289
17.	True Shared Memory Programming on SCI-based Clusters	
	Martin Schulz	291
17.1	Introduction	291
17.2	Designing a Global Virtual Memory	292
17.2.1	Building Block 1: SCI-based Hardware DSM	292
17.2.2	Building Block 2: Software DSM Systems	293
17.2.3	Combining Both Building Blocks to the SCI-VM	293
17.2.4	Locality Issues and Caching	294
17.3	SCI-VM Implementation Challenges	295
17.3.1	Mapping of Individual Page Frames	295
17.3.2	Dynamically Paged Memory	296
17.3.3	Enabling Caching Using Relaxed Consistency	296
17.4	Framework for SCI-VM-based Programming Models	297
17.4.1	SCI-VM Interface	297
17.4.2	Tradeoff Between Transparency and Performance	298
17.4.3	Current Status of the Framework	298
17.5	SPMD Programming Model on Top of SCI-VM	299
17.5.1	The Execution Model	299
17.5.2	Allocating Shared Memory	300

17.5.3	Synchronization	300
17.5.4	Consistency Model	301
17.6	Experiments and Results	302
17.6.1	Experimental Setup	302
17.6.2	Results for the Numerical Kernels	302
17.6.3	Results for the Volume Rendering Code	304
17.7	Using the SCI-VM for Transparent Multithreading.....	305
17.7.1	Transparent Thread Distribution	305
17.7.2	Synchronization Mechanisms	306
17.7.3	Applying a Relaxed Consistency Model	306
17.8	Related Work	307
17.9	Conclusions and Future Work	308
18.	Implementing a File System Interface to SCI	
	P.T. Koch, J.S. Hansen, E. Cecchet, X. Rousset de Pina	313
18.1	Introduction	313
18.1.1	Motivation	313
18.1.2	SCI-based File Systems	314
18.1.3	Outline	314
18.2	Sharing in File Systems	315
18.2.1	Memory-Mapped Files	315
18.2.2	UNIX Example with a Memory-Mapped File.....	316
18.2.3	File Consistency.....	316
18.2.4	Synchronization	317
18.3	Issues for Implementing SCI-based File Systems	317
18.3.1	A Virtual File System.....	318
18.3.2	Files and Directories	319
18.3.3	Example of Vnode/vfs Data Structures.....	319
18.3.4	Virtual File System Operations	320
18.3.5	Interaction with the Virtual Memory System.....	321
18.3.6	Remote Memory Mappings and File Consistency	322
18.3.7	Synchronization	322
18.4	The SciOS Prototype	323
18.4.1	SciOS Memory Protocols	323
18.4.2	Main File System Data Structures	324
18.4.3	The GLOBAL Memory Protocol	325
18.4.4	Memory Protocol Implementation in Linux	327
18.5	Related Work	328
18.6	Summary and Conclusions	329
19.	Programming SCI Clusters Using Parallel CORBA Objects	
	Thierry Priol, Christophe René, Guillaume Alléon.....	333
19.1	Introduction	333

19.2	Parallel vs. Distributed Programming	333
19.3	An Overview of CORBA	335
19.4	Parallel CORBA Objects	336
19.4.1	Execution Model	336
19.4.2	Extended-IDL	337
19.4.3	Implementation of Parallel CORBA Objects	340
19.5	The <i>Cobra</i> Runtime System	340
19.5.1	<i>Cobra</i> Services	341
19.5.2	<i>Cobra</i> Software Architecture	342
19.6	A Case Study: The IDAHO Application	344
19.7	Related Work	346
19.8	Conclusion and Perspectives	347
20.	The MuSE Runtime System for SCI Clusters: A Flexible Combination of On-Stack Execution and Work Stealing	
	Markus Leberecht	349
20.1	Introduction	349
20.2	The MuSE System	351
20.2.1	The SMiLE Cluster of PCs	351
20.2.2	The Multithreaded Scheduling Environment	352
20.3	Experimental Evaluation	357
20.3.1	Basic Runtime System Performance	357
20.3.2	Load Balancing and Parallelism Generation	358
20.4	Related Work and Conclusion	362
<hr/>		
Part VII. Benchmark Results and Application Experiences		
<hr/>		
21.	Large-Scale SCI Clusters in Practice: Architecture and Performance	
	Jens Simon, Alexander Reinefeld, Oliver Heinz	367
21.1	Introduction	367
21.2	PSC System Architecture	367
21.2.1	Node Configuration	368
21.2.2	SCI Interconnect	369
21.2.3	Software Configuration	370
21.3	Standard Benchmarks	371
21.3.1	Low-Level MPI Benchmarks	371
21.3.2	Parallel Linpack	373
21.3.3	FFT Benchmarks	374
21.3.4	HINT Benchmark	375
21.4	Applications	376
21.5	Summary	379

22. Shared Memory Parallelization of the GROMOS96 Molecular Dynamics Code

Marcus Dormanns	383
22.1 Introduction	383
22.2 The GROMOS Code	384
22.2.1 General Code Characteristics	384
22.2.2 Structure of the Code	384
22.3 Parallelization	386
22.3.1 Starting with Parallelism and Coordinating I/O	386
22.3.2 Parallelization of the Interaction Calculation Kernels ..	387
22.4 Performance Results	392
22.4.1 Hardware Platform	392
22.4.2 Results	392
22.4.3 Performance Comparison to Other Parallel GROMOS Implementations	393
22.5 Conclusion	394

23. SCI Prototyping for the Second Level Trigger System of the ATLAS Experiment

A. Belias, A. Bogaerts, D. Botterill, J. Dawson, E. Denes, F. Giacomini, R. Hauser, C. Hortnagl, R. Hughes-Jones, S. Kolya, D. Mercer, R. Middleton, J. Schlereth, P. Werner, F. Wickens	397
23.1 Introduction	397
23.2 The ATLAS Trigger System	397
23.3 Low-Level API	399
23.3.1 Basic Performance Measurements	400
23.4 The ATLAS Level-2 Trigger Demonstrator	403
23.4.1 Hardware	405
23.4.2 Software	406
23.4.3 Vertical Slice Configurations	407
23.4.4 Conclusions	410
23.5 Objectives and Design of the Second Prototype	410
23.5.1 Lessons Learned from the Demonstrator	410
23.5.2 Testbed	411
23.5.3 Software	413
23.5.4 SCI Testbed	413

Part VIII. Tools for SCI Clusters

24. SCI Monitoring Hardware and Software: Supporting Performance Evaluation and Debugging

Wolfgang Karl, Markus Leberecht, Michael Oberhuber 417

24.1 Introduction 417

24.2 The Monitoring Approach for the SMiLE PC Cluster 418

24.3 The Controlled Deterministic Execution Approach (*CODEX*) 422

24.4 Controlling Execution with SMiLE 425

24.4.1 Mapping POEM to the SMiLE Architecture 425

24.4.2 Controlling Execution on SMiLE 426

24.4.3 A Framework for an Implementation of *CODEX* for
Fine-Grained DSM Execution 428

24.5 Related Work 430

24.6 Conclusion 430

25. Monitoring SCI Clusters

Matthias Maier-Stahel, Roger Butenuth, Hans-Ulrich Heiss 433

25.1 Motivation 433

25.2 General Architecture 434

25.3 Monitor Agents 434

25.4 Master 436

25.5 Visualizer 437

25.6 Conclusion 440

26. Multi-User System Management on SCI Clusters

Matthias Brune, Axel Keller, Alexander Reinefeld 443

26.1 Introduction 443

26.1.1 Hardware Scenario 443

26.1.2 Software Scenario 444

26.1.3 User Access and System Management 445

26.2 Architecture of CCS 445

26.2.1 Island Concept 445

26.2.2 User Interface 446

26.2.3 Scheduling 448

26.2.4 Partitioning the System 450

26.2.5 Job Creation and Control 451

26.2.6 Reliability 453

26.3 Resource and Service Description 454

26.3.1 Graphical Representation 455

26.3.2 Textual Representation 456

26.3.3 Internal Data Representation 457

26.4 Related Work	459
26.5 Summary	459

Part IX. Perspectives

27. Industrial Takeup of SCI and Future Developments

David B. Gustavson	465
27.1 SCI's Cultural Context	465
27.2 SCI Marketing and Adoption	469
27.3 Commercial Adoption of SCI	472
27.3.1 Interface Chips and Products	472
27.3.2 Coherent Shared Memory Implementations	473
27.3.3 Non-coherent Implementations	475
27.4 Future Directions	476
27.4.1 IEEE P2100 (SerialPlus)	477
27.4.2 Concurrent Buses—A New Name for this Technology .	478
27.4.3 Concurrent Behavior is Essential for Scalability	479

List of Contributors	481
-----------------------------------	-----

Subject Index	487
----------------------------	-----

SCI: Scalable Coherent Interface
Architecture and Software for High-Performance
Compute Clusters

Hellwagner, H.; Reinefeld, A. (Eds.)

1999, XXII, 494 p., Softcover

ISBN: 978-3-540-66696-7