

# Preface

## Background

System interconnection networks have become a critical component of the computing technology of the late 1990s, and they are likely to have a great impact on the design, architecture, and use of future high-performance computers. Indeed, it is today not only the sheer computational speed that distinguishes high-performance computers from desktop systems, but the efficient integration of the computing nodes into tightly coupled multiprocessor systems. Network adapters, switches, and device driver software are increasingly becoming performance-critical components in modern supercomputers.

Due to the recent availability of fast commodity network adapter cards and switches, tightly integrated clusters of PCs or workstations have emerged on the market, now filling the gap between desktop systems and supercomputers. The use of commercial off-the-shelf (COTS) technology for both computing and networking enables scalable computing at relatively low costs. Some may disagree, but even the world champion in high-performance computing, Sandia Lab's *ASCI Red* machine, may be seen as a COTS system. With just one hardware upgrade (pertaining to the Intel processors, not the network), this system has constantly been number one in the TOP-500 list of the worldwide fastest supercomputers since its installation in 1997. Clearly, the system area network plays a decisive role in overall performance.

The Scalable Coherent Interface (SCI, ANSI/IEEE Standard 1596-1992) specifies one such fast system interconnect, emphasizing the flexibility, scalability, and high performance of the network. In recent years, SCI has become an innovative and widely discussed approach to interconnecting multiple processing nodes in various ways. SCI's flexibility stems mainly from its communication protocols: in contrast to many other interconnects, SCI is not restricted to either message-based or shared-memory communication models. Instead, it combines both, taking advantage of similar properties that have been investigated in such hybrid machines as Stanford's FLASH or MIT's Alewife architectures. Since SCI also defines a distributed directory-based cache coherence protocol, it is up to the computer architect to choose from a broad range of communication and execution models, including efficient message-passing architectures, as well as shared-memory models, in either the NUMA or CC-NUMA variants.

European industry and research institutions have played a key role in the SCI standardization process. Based on SCI adapter cards, switches, and fully integrated cluster systems manufactured by European companies, the SCI community in Europe has made and is making significant developments and state-of-the-art research on this important interconnect.

## **Purpose of the Book**

From many discussions with friends, colleagues, and potential users, we found that one significant barrier to the widespread deployment and use of SCI is the lack of a clear vision of how SCI works, how it is being used in building clusters, and how obstacles in its deployment can be avoided. Our goal in compiling this book is to address these barriers by providing in-depth information on the technology and applications of SCI from various perspectives. The book focuses on SCI clusters built from commodity PCs or workstations and SCI adapters, since they represent the mainstream and most cost-effective application of SCI to date.

In addition, some challenging research issues, mostly pertaining to shared-memory programming on SCI clusters, are discussed and potential improvements for SCI cluster equipment are highlighted.

Who is the intended audience? The relevance of the book for computer architects is obvious, given the importance of system area networks for modern high-performance computers. But the book is also intended for system administrators and compute center managers who plan to invest in cluster technology with COTS components. Furthermore, researchers and students wanting to contribute to this interesting technology with their own hard- or software developments might find this book helpful.

## **Organization of the Book**

The book consists of nine parts, each subdivided into chapters covering individual topics. On the whole, the contributions cover the complete hardware/software spectrum of SCI clusters, ranging from the major concepts of SCI, through SCI hardware, networking, and low-level software issues, various programming models and environments, up to tools and application experiences.

Part I introduces the SCI standard and its application in practical computer systems. SCI is put into context by comparing its concepts, architecture, and performance with its strongest competitor Myrinet and also with the proprietary Cray T3D interconnection network which has set the standards back in 1993.

Part II looks at the hardware. It describes two implementations of SCI adapters, the commercial, widely used Dolphin SCI cards for the PCI and SBus I/O buses, and the prototype adapter developed at TU München which can be extended by special hardware for monitoring the SCI packet flow.

Building on the hardware, Part III explores how to build SCI interconnection networks and analyzes various critical aspects of SCI networks, among them ringlet scalability and potential performance degradation by hardware-generated retry traffic.

Part IV moves on to software, describing the functionality and concrete implementations of SCI device drivers and introducing a low-level API that abstracts away SCI's distributed shared memory (DSM) implementation details from higher-level software.

The first class of parallel and distributed programming models, namely message-passing libraries on top of SCI, are covered in Part V. The chapters report on projects which implemented sockets, TCP/IP, PVM, and MPI with high efficiency on top of SCI, by making judicious use of the SCI DSM and related features.

As pointed out by the contributions in Part VI, developing shared-memory programming environments on SCI clusters with current SCI hardware and driver software is more challenging than implementing message-passing libraries. Partly due to the lack of well established shared-memory standards, the approaches described are widely diverse. They range from specific shared virtual memory systems on top of SCI to a fully transparent, distributed thread system and to shared, parallel objects extending a CORBA middleware implementation. The chapters discuss some of the limitations of current SCI cluster equipment and present potential routes for future developments.

Real-world experiences with SCI clusters are reported in Part VII. As a reference, benchmark and application performance results from the very large SCI clusters that are operated at PC<sup>2</sup> Paderborn are given first. The parallelization approaches and performance results from two projects, a complex molecular dynamics code and a real-time data acquisition and filtering application prototype for high-energy physics, are described as examples of real-world uses of SCI clusters.

Part VIII deals with tools for SCI clusters, which apparently are still in their infancy. Therefore, only two basic SCI monitors, one implemented in hardware, the other in software, and their potential applications are presented here. In addition, a powerful system management tool, developed to operate the large Paderborn clusters as general-purpose, multi-user compute servers is introduced.

Both SCI and SCI interconnects are still evolving in terms of standardization, product development, research findings, and applications. In the final part, Part IX, therefore, one of the designers of SCI, David Gustavson, describes the perspectives that he sees for SCI.

## Acknowledgements

With great pleasure, we acknowledge the efforts of the many individuals who have contributed to the development of this book. First and foremost, we thank the authors for their enthusiasm, time, and expertise which made this

book possible. We are also grateful to the people who helped in organizing the book, especially Oliver Heinz (PC<sup>2</sup> Paderborn), Hans-Hermann Frese (ZIB Berlin), and Angelika Rossak (University Klagenfurt). The European Commission provided financial support through the ESPRIT IV Programme's SCI Working Group (EP 22582). Finally, we acknowledge the help of Alfred Hofmann and Antje Endemann of Springer-Verlag, who were always competent, professional, and efficient partners to work with.

September 1999

*Hermann Hellwagner*  
*Alexander Reinefeld*

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