SMiLE project, further work on interconnection networks is planned to identify and analyze suitable topologies for large SCI-based parallel machines, for instance.

The group has cooperations with several US universities (Emory University, Atlanta, GA; Southern Methodist University, Dallas, TX; Santa Clara University, Santa Clara, CA) and close links to European companies and institutions engaged in SCI-related projects (e.g. Dolphin Interconnect Solutions, Oslo; CERN, Geneva; Parallab at University of Bergen). Joint projects are currently being prepared.

Finally, it must be noted that the majority of the researchers in this group are heavily involved in teaching as well.

4.1 SMiLE—Shared Memory in a LAN-like Environment

4.1.1 The SMiLE Project

by Hermann Hellwagner, Wolfgang Karl, and Harald Richter

The SMiLE project is a joint effort of the architecture group to set up and to develop software for a low-cost, yet technically advanced and powerful parallel computing resource. The system is to consist of standard desktop computers (PCs based on Pentium and the PCI bus) and a standard interconnect, the Scalable Coherent Interface (SCI) [Gus92] [IEE93]. Based on SCI, the machine will be a shared memory multiprocessor built from distributed computing resources, which yielded the acronym for the project: *SMiLE*. Other researchers use the term *Local-Area MultiProcessor (LAMP)* for this type of parallel machine [GL94].

An outline of the project is given in the sequel. Individual contributions to SMiLE are described in the rest of this chapter.

4.1.1.1 Motivation

There are two major reasons behind the SMiLE project:

- the demand for low-cost parallel processing, and
- the potential of SCI and SCI-based local-area multiprocessors.

Networks of workstations (NOWs) have become increasingly popular as low-cost and widely available facilities for parallel processing. Public domain packages like PVM, p4, and NXLib, for instance, allow a NOW to be utilized as a virtual parallel computer at practically no costs. This opportunity is exploited by small and medium enterprises, universities, and research laboratories which do have demand for parallel computing for various applications, research projects, and teaching.

Parallel computing with NOWs has two drawbacks, however: insufficient communications performance over today's LANs (still typically Ethernet), and message-passing programming only. SCI and SCI-based distributed shared memory (DSM) multiprocessors have the potential to alleviate these problems.

SCI is defined to provide 1 Gbit/s or 1 Gbyte/s bandwidth, while offering the chance for low-latency communications software through specific SCI hardware protocol features, e.g. guaranteed data delivery. This has been demonstrated for instance in the Convex Exemplar SPP 1000/1200 parallel machines that use SCI rings for inter-hypernode communications. With commercial off-the-shelf SCI products from Dolphin Interconnect Solutions, Oslo, Norway, application-level communication latencies below 10 μs in an SCI-based cluster of SPARCstations-20 have been demonstrated [Dol95].

Although based on point-to-point communication links among nodes, SCI offers all the standard services of a computer bus. Besides message-passing transactions, the standard defines transactions and protocols for (coherent or non-coherent) memory accesses as well as atomic synchronization primitives. Therefore, given a full SCI implementation, a SCI-based cluster may be programmed using traditional message-passing styles; in addition, it may be employed as a (distributed) shared memory parallel machine, with more convenient programming models being able to be implemented.

4.1.1.2 Objectives and Current Work

With this background, the SMiLE project commenced in 1995. The general objectives as well as ongoing work are outlined in the following. The reader is referred to more detailled descriptions of the contributions in the rest of the chapter.

Development of a SCI-based PC cluster with DSM

A low-cost DSM multiprocessor using standard PCs (based on Pentium and the PCI bus) interconnected by SCI technology, will be built up and demonstrated. Commercial SCI products will be used as building blocks as far as possible. However, since e.g. PCI-SCI interface cards are not yet available, we started our own hardware and system software developments.

Current work in this hardware-oriented area comprises the following:

- Design of a PCI-SCI interface card (see subsection 4.2.1 on page 114).
  Although Dolphin and CERN have announced PCI-SCI adapter cards, we commenced our own PCI-SCI interface design using VHDL-based design and simulation tools. The motivation for this work is that the performance analysis tool PATOP and the debugging tool DETOP, described earlier in this report, are desirable to be ported to and extended for an SCI platform, and be supported by monitoring hardware. Our own development gives us the chance to design and integrate monitoring "hooks" into the interface board.

- Performance analysis of SCI interface board designs and SCI cluster configurations (see subsection 4.2.2 on page 116)

Software development for SCI-based multiprocessors

While SCI hardware is slowly becoming available on the market, software support is lagging behind. We aim to provide a software basis for parallel processing on the SMiLE multiprocessor to be developed in the first area. Porting and adapting both an existing message-oriented parallel programming package (PVM), and a shared memory-oriented programming environment, e.g. a thread package, are envisaged. The principal objectives behind these porting efforts are to exploit the performance potential of the SCI cluster and to enable more convenient parallel programming in a cluster environment than possible with message passing. Furthermore, port and adaption of the parallel program development tools PATOP and DETOP to this machine are planned.

Initial work carried out in the software area is:

- Adaption of PVM to transfer data over native SCI (see subsection 4.1.2).
  The goal of this work is to provide efficient message passing on PVM application level by directly implementing PVM data transfers on native SCI communication mechanisms. The current platform for this work is a two-node workstation "cluster" with first-generation SCI products. Initial experiences with this testbed, the current implementation, and performance results are given in some detail below.
- Port of a Unix Shared Virtual Memory (SVM) implementation onto the SCI testbed (see subsection 4.1.2)

Research into efficient parallel programming of DSM systems

In the long term, the SCI multiprocessor will provide a vehicle for experimental research into efficient use of a DSM parallel machine. The key to efficiency is to provide and exploit data locality in order to avoid remote memory access latencies as far as possible. It is envisaged to study and develop mechanisms that monitor data access behaviour at run time, and move or replicate data and migrate processes to improve locality of data accesses and, thus, to increase efficiency of a DSM machine.

An alternative approach using latency hiding is currently being pursued in the design of a novel execution model based on the dataflow paradigm for the SMiLE architecture:

- Multithreading Scheduling Environment (MuSE) (see subsection 4.2.3 on page 116).
  The basic concept is to exploit a specific feature of the SMiLE architecture, low-latency *buffered writes*, to implement a highly efficient runtime multithreading system for the dataflow-like execution of programs on a parallel system with conventional microprocessors, e.g. the Pentium. Threads are derived by partitioning a dataflow graph generated from functional programs (written in SISAL), and executed when their input data are available. The main idea is not to use read operations to fetch input data into a thread, but to transfer data among interdependent threads using buffered writes only. On our current testbed, a buffered (remote) write has an order of magnitude lower latency than a (remote) read operation. The time gained using this approach can be used for the synchronization and scheduling mechanisms required for the dataflow-like programs. Initial analyses indicate that this approach can result in highly efficient execution on the SMiLE DSM machine; see 4.2.3 on page 116 for details.