SCI-VM: A Flexible Base for Transparent Shared Memory Programming Models on Clusters of PCs

Martin Schulz
schulzm@in.tum.de
Lehrstuhl für Rechnertechnik und Rechnerorganisation, LRR–TUM
Institut für Informatik, Technische Universität München

Abstract. Clusters of PCs are traditionally programmed using the message passing paradigm as this is directly supported by their loosely coupled architecture. Shared memory programming is mostly neglected although it is commonly seen as the easier and more intuitive way of parallel programming. Based on the user-level remote memory capabilities of the Scalable Coherent Interface, this paper presents the concept of the SCI Virtual Memory which allows a cluster-wide virtual memory abstraction. This SCI Virtual Memory offers a flexible basis for a large variety of shared memory programming models which will be demonstrated in this paper based on an SPMD model.

1 Introduction

Due to their excellent price–performance, clusters built out of commodity–off–the–shelf PCs and connected with new low–latency interconnection networks are becoming increasingly commonplace and are even starting to replace traditional massively parallel systems. According to their loosely coupled architecture, they are traditionally programmed using the message passing paradigm. This trend was further supported by the wide availability of high–level message passing libraries like PVM and MPI and intensive research in low–latency, user–level messaging [17, 12].

Next to the message passing paradigm, which relies on explicit data distribution and communication, a second parallel programming paradigm exists, the shared memory paradigm. This paradigm, which offers a global virtual address space for sharing data between processes or threads, is preferably utilized in tightly coupled machines like SMPs that offer special hardware support for a globally shared virtual address space. It is generally seen as the easier programming model, especially for programmers that are not used to parallel programming, but it comes with the price of higher implementation complexity either in the form of special hardware support or in the form of complex software layers. This prohibits their widespread use on cluster architectures as the necessary hardware support is generally missing.

In order to overcome this gap and provide a bridge between tightly and loosely coupled machines, we utilize the Scalable Coherent Interface (SCI) [3,
as the interconnection technology. This network fabric, which is standardized in the IEEE standard 1596–1992 [14], utilizes a state of the art split transaction protocol and currently offers bandwidths of up to 400 MB/s. The base topology for SCI networks are small ringlets of up to 8 nodes which hierarchically can be organized to configurations of up to 64K nodes.

The core feature of SCI based networks is the ability to perform remote memory operations through direct hardware distributed shared memory (DSM) support. Figure 1 gives a general overview of how this capability can be applied. The basis is formed by the SCI physical address space which allows addressing of any physical memory location on any connected node through a 64 bit identifier (16 bit to specify the node, 48 bit to specify the physical address). From this global address space, each node can import pieces of remote memory into a designated address window within the PCI address space using special address translation tables on the SCI adapter cards. After mapping the PCI address space into the virtual memory of a process, the remote memory can be directly accessed using standard user-level read and write operations. The SCI hardware forwards these operations transparently to the remote node and, in case of a read operation, returns the result. Due to the pure hardware implementation avoiding any software overhead, extremely low latencies of about 2.3 µs (one way) can be achieved.

![Fig. 1. Mapping remote memory via SCI from node A into the virtual address space of process j on node B.](image)

Unfortunately, this hardware DSM mechanism can not directly be utilized to construct a true shared memory programming model as it is only based on sharing physical memory. It misses an integration into the virtual memory management of the underlying operating system. Extra software mechanisms have to be applied to overcome this limitation and to build a fully transparent global virtual address space. The SCI Virtual Memory layer, presented in this paper, implements these concepts on a Windows NT platform forming the basis for any kind of true shared memory programming on clusters with hardware DSM support.