

The MPI Implementation

Introduction

Nowadays MPI is the state of the art message passing paradigm. To achieve a well balanced, competitive communication subsystem, the hardware and software design must be done together. Therefore one of the major design goals of the T-NET network was the support of an efficient MPI implementation.

Developing a new MPI Implementation

Before 1993 each vendor provided his own communication language. In the year 1993 vendors and users founded the MPI Forum to standardise the message passing paradigms. In 1994 the standard version 1.0 was released. Minor modifications followed 1995 with the standard 1.1 and in 1997 some corrections led to the standard 1.2.

The implementation of the FCI library was started 1997 on a bus based prototype of the T-NET hardware. The experiences of that design led to the implementation of the current switch based T-NET hardware. The ADI and MPI library itself were developed 1995 and 1996 for a previous system called "GigaBooster".

The Principle Behind

The design goals of the FCI/MPI implementation were:

- Scaling up to thousands of processes
- Low latency (less than 30 μ s)
- High bandwidth (PCI bandwidth)
- Good performance for typical MPP applications
- Most efficient usage of system resources
- cost effective system solution

To achieve a low latency, the operating system must be by-passed. The communication hardware can directly be accessed by the user application. The tasks of the operating system such as security checking or virtual to physical address translation is rebuild in the communication hardware.

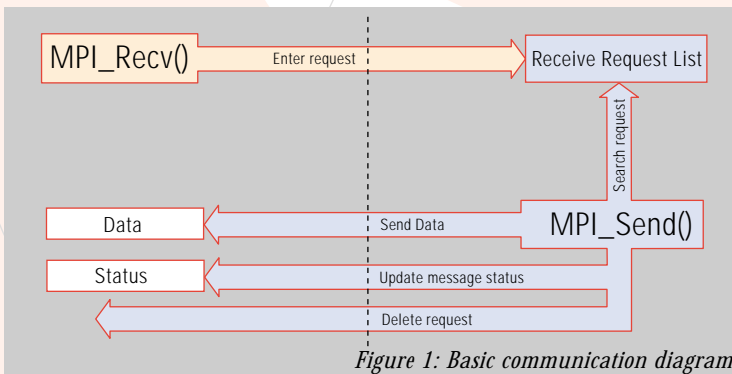


Figure 1: Basic communication diagram

High bandwidth can only be achieved by avoiding any buffering (zero copy). FCI/MPI has no internal buffers, the data is directly transferred from application to application memory.

For good scaling an efficient flow control is needed in order to avoid bottlenecks and hot spots in the communication network. The FCI library enters a receive request in the remote receive request list. Each send

searches that list till it has found a matching receive request and sends the data (see figure 1).

Standard compliance

The FCI/MPI implementation is compliant to the MPI standard 1.2.

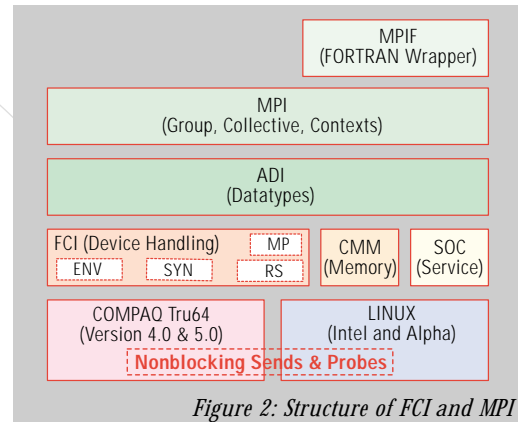


Figure 2: Structure of FCI and MPI

Structure of Libraries

Figure 2 shows the structure of the FCI and MPI libraries. The device drivers map the communication hardware into the application memory. Non-blocking sends and probes are also executed in device driver context. The FCI (Fast Communication Interface) library provides one-sided and message passing communication. The ADI (Abstract Device Interface) library builds MPI datatypes on top of FCI. Collective operations and communicator handling is implemented in the MPI library. The MPIF library provides a FORTRAN interface to the MPI library.

Verification of MPI Implementation

The MPI implementation was verified with the INTEL/DARPA MPI validation suite. This suite consists of over 300 different test programs covering the whole MPI functionality.

Benchmark Results

On the T-NET hardware, FCI/MPI achieves a half-round trip latency for a ping-pong benchmark of 20 μ s. A bandwidth of 72 MB/s was measured.

Future Development Road map

Currently an optimisation of multicast communications is in development. This will allow the usage of network multicast for collective operations. One-sided communication will be implemented on FCI level and a SHMEM-like interface built on top of it. Support for the Totalview debugger is under development.

An MPI-IO interface will be available in 3rd quarter 2000.

Contact Person

Martin Frey
e-mail: martin.frey@scs.ch
Phone: +41 (0)1 445 16 00