MPI in 2002: Has it Been Ten Years Already?

Ewing Lusk
Mathematics and Computer Science Division
Argonne National Laboratory

Abstract

In April of 1992, a group of parallel computing vendors, computer science researchers, and application scientists met at a one-day workshop and agreed to cooperate on the development of a community standard for the message-passing model of parallel computing. The MPI Forum that eventually emerged from that workshop became a model of how a broad community could work together to improve an important component of the high performance computing environment. The Message Passing Interface (MPI) definition that resulted from this effort has been widely adopted and implemented, and is now virtually synonymous with the message-passing model itself.

MPI not only standardized existing practice in the service of making applications portable in the rapidly changing world of parallel computing, but also consolidated research advances into novel features that extended existing practice and have proven useful in developing a new generation of applications.

This talk will discuss some of the procedures and approaches of the MPI Forum that led to MPI’s early adoption, and then describe some of the features that have led to its persistence as a reference model for parallel computing. Although clusters were only just emerging as a significant parallel computing production platform as MPI was being defined, MPI has proven to be a useful way of programming them for high performance, and we will discuss the current situation in MPI implementations for clusters.

MPI was deliberately designed to grant considerable flexibility to implementors, and thus provides a useful framework for implementation research. Successful implementation techniques within the MPI standard can be utilized immediately by applications already using MPI, thus providing an unusually fast path from research results to their application. At Argonne National Laboratory we have been developing and distributing MPICH, a portable, high performance implementation of MPI, from the very beginning of the MPI effort. We will describe MPICH-2, a completely new version of MPICH just being released. We will present some of its novel design features that we hope will stimulate both further research and a new generation of complete MPI-2 implementations, along with some early performance results. We will conclude with a speculative look at the future of MPI, including its role in other programming approaches, fault tolerance, and its applicability to advanced architectures.