Open MPI on Mac OS X: Enabling big science on the Mac

Timothy I. Mattox, Ph.D. timattox@open-mpi.org
Open Systems Lab, Pervasive Technology Labs at Indiana University

http://www.open-mpi.org/

Abstract

Open MPI is a high performance implementation of the Message Passing Interface (MPI) developed through a collaboration of universities, national laboratories, and industry. MPI is a critical part of modern scientific computing, providing an interface for parallel computing that can be utilized on everything from dual processor laptops up to tens of thousands of processors on the fastest supercomputers in the world. We have designed Open MPI to scale across the entire spectrum of available configurations.

Open MPI provides a number of useful features on Mac OS X, including integration with XGrid and support for Universal Binaries. Applications can be run across a combination of Power PC and Intel Macs, allowing scientists to effectively use all their Mac computing resources.

For scientists with large computational needs, Open MPI supports communication over InfiniBand and Myrinet, as well as over TCP/IP and Gigabit Ethernet. For operation in large compute clusters, Open MPI integrates with traditional batch schedulers like PBS/Torque and SLURM. By using a single MPI implementation that supports a large number of platforms, scientists are able to spend less time working customizing their application to a particular MPI implementation's behavior and more time doing real science.

Introduction

Modern scientific research involves not only lab experimentation, but computer simulations and data processing. Simulations of events that can not be easily studied in a laboratory setting, such as protein folding or the spread of epidemic causing diseases, can allow discoveries that would otherwise be tedious or impossible. Lab experiments frequently result in large quantities of raw data, which must be processed before it can be usefully visualized and understood.

Frequently, these simulations and data processing tasks are so complicated that they require specialized computing infrastructure. The high performance computing (HPC) field is dedicated to developing large systems that meet these intense computing requirements. For computationally intensive tasks, most high performance computing needs are met with clusters of commodity hardware. The clusters tie a large number of individual machines together with software, generally using an interface called the Message Passing Interface, or MPI [3].

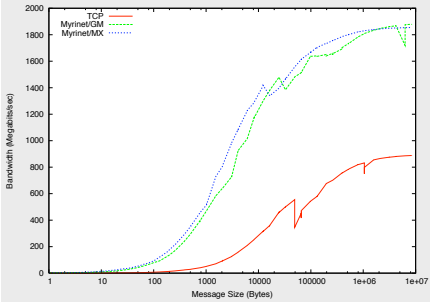
Open MPI, a new implementation of the MPI standard, is the result of a collaboration between universities, commercial HPC vendors, and U.S. national laboratories. A low overhead component architecture [2] allows Open MPI to be customized for a particular environment, allowing it to operate efficiently on everything from single processor laptops to the largest supercomputers in the world. The same component architecture also allows Open MPI to be customized to take advantage of features specific to the Mac OS X environment.

Acknowledgments

Project support was provided through the United States Department of Energy, National Nuclear Security Administration's ASCI/PSE program and the Los Alamos Computer Science Institute; a grant from the Lilly Endowment and National Science Foundation grants NSF-0116050, EIA-0202048 and ANI-0330620.

Performance

MPI implementations are generally measured by point-to-point performance, and Open MPI provides excellent performance in this realm. The graph below presents the performance of Open MPI between two G5 XServe machines. Both machines contain two 2.3 GHz G5 processors and 4 GB of memory. The Myrinet cards are LANai 10, PCIX-D cards directly connected. A single Gigabit Ethernet connection to a Bay Stack switch also connects the machines.



High speed interconnects such as Myrinet and InfiniBand require memory to be "prepared" before it can be used for communication. Because the network card writes incoming data directly into user memory, bypassing the kernel, physical pages can not be "moved" once communication has started. The cost of "pinning" the page so that it can be used for communication is frequently higher than the cost of sending or receiving data in that page. Traditional solutions to this problem require hooks into the malloc system in libc and, on Mac OS X, often require forcing a flat name-space. Even with the malloc system hooks, best performance is only realized if buffer reuse is extremely high. Open MPI takes a unique solution to the problem, using a communication pipeline protocol that provides high performance without memory hooks, even when communication buffers are not reused.

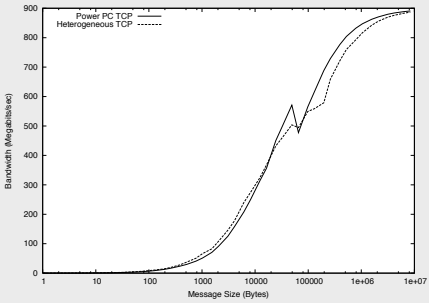
Broad Industry Support

Open MPI is used extensively in the HPC industry from small development clusters to some of the largest and fastest supercomputers in the world. Open MPI is available under the New BSD license, and thus gains from the Open Source community development model. Open MPI is developed by a core group consisting of the following fourteen organizations in alphabetical order:

- Cisco Systems, Inc.
- University of Houston
- High Performance Computing Center Stuttgart (HLRS)
- IBM
- Indiana University
- Los Alamos National Lab
- Mellanox Technologies
- Myricom, Inc.
- Oak Ridge National Laboratory
- QLogic Corporation
- Sun Microsystems
- Technische Universitaet Dresden
- University of Tennessee
- Voltaire

Heterogeneous Clusters

With the introduction of the Intel processor to the Apple family, heterogeneous computing has come to Mac OS X. Open MPI's data-type engine is designed to provide high performance, even when data must be converted between processor formats. The graph below shows the impact on bandwidth when integers are sent between Intel and Power PC machines, using TCP. While there is a slight performance impact for large messages, there is virtually no impact on small messages. This is because Open MPI hides the memory fix-ups as part of copying the message from internal memory to the user's message buffer.



Customized for Mac OS X

Mac OS X provides a number of features unique in HPC environments. The G5 processor and the Intel Core Duo processors both provide excellent performance for numerical applications. With Mac OS X 10.4 and XGrid, idle workstations can quickly be converted into a cluster for parallel computation. The Unix history of Mac OS X allows code developed on a Mac laptop or desktop to easily be moved to large institution supercomputers, even if they aren't running Mac OS X. We should know – Open MPI was largely developed on Mac laptops.

We have adapted Open MPI to take advantage of a number of features of the Mac OS X environment:

- Integration with XGrid
- Universal Binary support
- Support for heterogeneous environments
- Stack trace display during fatal errors

In addition, Open MPI contains a number of features that, while not Mac OS X specific, are useful in the Mac environment:

- Support for common networks:
  - + TCP/IP
  - + Shared memory
  - + InfiniBand (MVAPI and OpenIB)
  - + Myrinet (GM and MX)
- Multiple network device support
- PBS/Pro / Torque scheduler support
- Sun Grid Engine batch scheduler support
- Complete Fortran 90 bindings

Upcoming Open MPI versions will also support the uDAPL network programming interface.

New Research

While Open MPI is a production quality project, it is also an ideal base for research into HPC computing. A number of ongoing research projects are investigating how to improve performance for scientific applications on the latest cluster environments. Current research areas include multi-core optimizations, collectives performance enhancements, and one-sided communication protocols.

Multi-core Optimization
Processor affinity, memory affinity and process mapping are areas that have great potential for improving the performance of HPC applications on the next generation of parallel computers built with multi-core processors.

Collectives Enhancements
Multi-core processors and multi-socket computers require advanced collectives algorithms, such as broadcast and gather for peak performance. Significant work on hierarchical collectives has shown significant benefit [4]. We are extending this work by developing a tool to allow system administrators to customize point-to-point collective routines for a particular cluster, something that has shown great promise in initial work [2].

One-sided Communication
Applications with random communication patterns frequently performing poorly using traditional MPI calls. The MPI-2 One-sided communication chapter provides a one-sided option, but limitations in the standard limit performance. We are investigating extending the MPI interface to provide applications with the performance of pure one-sided interfaces, while providing interoperability with the rest of the MPI interface and support for commodity networks.

Conclusions

Mac OS X provides an ideal platform for high performance computing due to its ease of use and powerful processors. Open MPI has been customized to perform well on Mac OS X, including in heterogeneous situations. The component architecture of Open MPI also allows us easily to take advantage of Mac OS X-specific features, such as the XGrid platform.

References

[1] B. Barrett, J. M. Squyres, A. Lumsdaine, R. L. Graham, and G. Bosilca. Analysis of the Component Architecture Overhead in Open MPI. In Proceedings, 12th European PVM/MPI Users' Group Meeting, Sorrento, Italy, September 2005.

[2] Fagg, G., Pjesivac-Grbovic, J., Bosilca, G., Angskun, T., Dongarra, J. "Flexible collective communication tuning architecture applied to Open MPI," 2006 Euro PVM/MPI (submitted), Bonn, Germany, September, 2006.

[3] Message Passing Interface Forum. MPI: A Message Passing Interface. In Proc. of Supercomputing '93, pages 878-883. IEEE Computer Society Press. November 1993.

[4] Thilo Kielmann and Rutger F.H. Hofman and Henri E. Bal and Aske Plaat and Raoul A. F. Bhoedjang. MagPIe: MPI's collective communication operations for clustered wide area systems. In ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'99), 34(8), pp131-140, May 1999.