The Local Area Multicomputer (LAM) Implementation of MPI

Jeffrey M. Squyres, Andrew Lumsdaine

Department of Computer Science and Engineering

University of Notre Dame
Overview

- What is LAM/MPI?
- Why would I use LAM/MPI?
- How do I use LAM/MPI?
- Where do I get LAM/MPI?
- Future directions
What is LAM/MPI?

- An independent implementation of the MPI standard
- All of MPI-1 (except MPI_CANCEL sent messages)
- Much of MPI-2
- Originally developed at the Ohio Supercomputing Center
  - Now developed / maintained at the University of Notre Dame
MPI-2 Features

- Dynamic processes
- Most of one-sided communication
- Most new MPI-2 datatypes
- Many MPI-2 support functions
- MPI-IO (from the ROMIO package)
- C++ bindings for MPI-1 functions
- Interoperable MPI (IMPI) point-to-point support
Usability Features

- Persistent, daemon-based run-time environment
- Visual debugging through XMPI
- Supports SPMD and MPMD execution models
- Pseudo-tty support (i.e., line-buffered output)
- Can `mpirun` debuggers / scripts
- Can be used with Purify and other memory-checking tools
- *Lots* of documentation
"Cluster Friendly"

- Guaranteed cleanup of user (runaway) processes
- Fast `mpirun` startup, even across large numbers of hosts
- SMP-aware `mpirun` syntax
- Passing of environment to remote ranks
- Works even in non-uniform filesystem environments
- POSIX-like path semantics
Supported Architectures

- Works on just about all POSIX architectures
  - Does not work under Windows
  - ...except under Cygwin
- Supports heterogeneous environments
- 64-bit clean
Overview

- What is LAM/MPI?
- Why would I use LAM/MPI?
- How do I use LAM/MPI?
- Where do I get LAM/MPI?
- Future directions
Why would I use LAM/MPI?

- It’s free!
- Under continual development
  - We’re not just the developers, we’re users too
  - New research directions lead to better performance
- Bunches of MPI-2 already implemented
- Ability to `mpirun` scripts and arbitrary debuggers
  - We use this feature extensively to develop LAM itself
Cluster Friendliness

- Integrated process management
- Integrated fast `mpirun` startup
Even More Reasons

- High performance
  - Transparent dual mode shared memory / TCP message passing
  - Optimized common-case send/receive
  - Optimized persistent mode send/receive
- POSIX behavior for serial-like execution semantics
XMPI

- Visualization of message passing
Overview

- What is LAM/MPI?
- Why would I use LAM/MPI?
- How do I use LAM/MPI?
- Where do I get LAM/MPI?
- Future directions
How do I use LAM/MPI?

- Three main steps:
  1. Start the LAM/MPI run-time environment
     
     `lamboot -v hostfile`

  2. Run user program(s)
     
     `mpirun -np 4 program1`
     `mpirun -np 8 program2`
     `...`

  3. Shutdown the LAM/MPI run-time environment
     
     `lamhalt`
Compiling

- “Wrapper” compilers take care of all necessary flags
- Used just like “real” compilers

\[ \text{C: } \text{mpicc } \text{foo.c} \]
\[ \text{C++: } \text{mpiCC } \text{bar.cc} \]
\[ \text{Fortran: } \text{mpif77} \text{ baz.f} \]

- Can change the underlying compiler

\[ \text{setenv LAMHF77 f90} \]
\[ \text{mpif77 baz.f} \]
Process Management

- **lamclean**: Clean up “runaway” processes
  - Most helpful when debugging parallel code
  - Especially if \^C, for some reason, doesn’t kill everything

- **mpitask**: Check progress of MPI ranks

```
TASK    FUNC   SRC    TAG    COMM   CNT   DTYPE
0/0  a.out  Recv  1/1   1234    WORLD 1024 INT
```

- **mpimsg**: See pending messages on the network

```
SRC    DEST   TAG    COMM   CNT   DTYPE
1/1  0/0   4321    WORLD 1024 INT
```
Overview

- What is LAM/MPI?
- Why would I use LAM/MPI?
- How do I use LAM/MPI?
- Where do I get LAM/MPI?
- Future directions
Where do I get LAM/MPI?

- The main LAM web site is:
  
  http://www.mpi.nd.edu/lam/

- Also contained in the leading Linux and BSD distributions
  
  - RedHat
  - S.u.S.E.
  - Debian
  - BlackLab
  - LinuxPPC
  - OpenBSD
Additional Information / Documentation

- The LAM FAQ contains much information about LAM, MPI, and typical cluster-based setups
  
  http://www.mpi.nd.edu/lam/faq/

- The LAM mailing list archives

  http://www.mpi.nd.edu/MailArchives/lam/

- To join the LAM mailing list, send mail to
  
  “majordomo@mpi.nd.edu” with “subscribe lam” in the body
Overview

- What is LAM/MPI?
- Why would I use LAM/MPI?
- How do I use LAM/MPI?
- Where do I get LAM/MPI?
- Future directions
Future Directions

- Myrinet and VIA “drivers”
- Tighter integration with PBS
- Improved shared memory performance
- Full IMPI functionality
- TotalView debugger support
- Thread safety / thread concurrency
- C++ bindings for MPI-2 functions
- Fortran 90 module