MPI ABI Working Group status

March 10, 2008

Jeff Brown, chair (LANL)

Motivation

Current MPI spec is an API, which allows for source but not binary compatibility across MPI implementations

Applications built on MPI must recompile for every different target MPI implementation

Not a problem on vertically integrated vendor systems with one MPI implementation

A pain on open systems (e.g. linux clusters) with multiple MPI offerings (mpich, lampi, openmpi, mvapich, …)

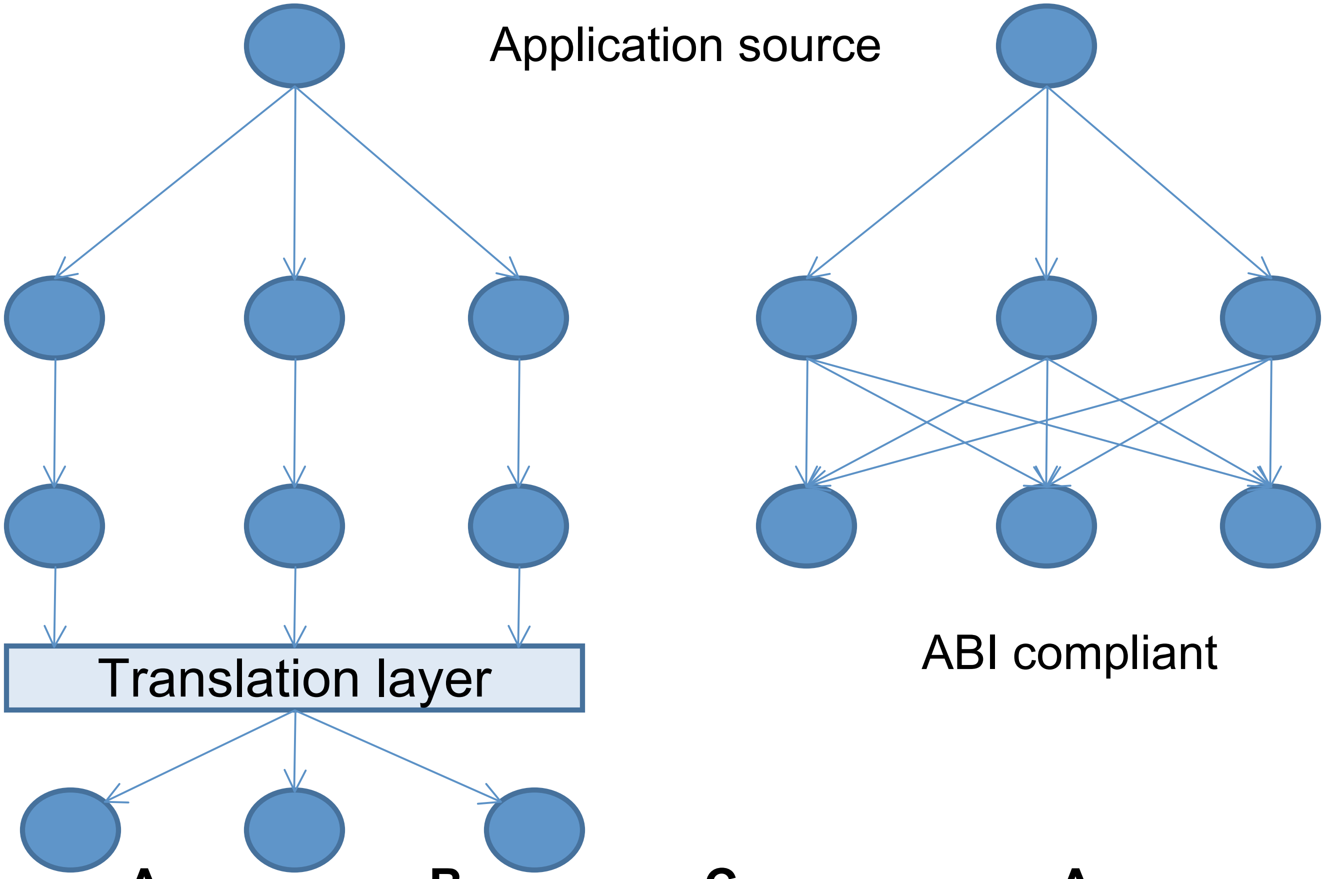
Current MPI situation

What we want

Compile time (API)

Link time (ABI)

Run time



Application source

Translation layer

ABI compliant

MPI Implementation A B C A
B C

Impacts

ISVs with products that use MPI (e.g. Matlab) must build and ship a separate binary for every target MPI implementation

Users cannot run third party applications distributed in binary form with an MPI implementation optimized for their hardware

Application code developers (like at LANL) must recompile for every different MPI

Tool providers (e.g. Tau) that use the profiling interface must build for every MPI

Adds to the complexity of system maintenance due to additional run-time dependencies (modules, etc.)

MPI ABI Initiative

Define an Application Binary Interface standard for MPI which will allow run-time dynamic linking to any compliant MPI binary library on a target platform (practical linkage compatibility)

Initial focus will be on the C bindings only

Not dealing with fortran issues (e.g. symbol names) for now (maybe never)

Not dealing with run time environment issues such as standard library names, paths, etc. – leave this to the sys admins to standardize to their local conventions

Does not ensure run-time correctness

Translation (or morph) layers may be required to allow for backward compatibility with binaries compiled with a non-ABI compliant MPI

What we need to standardize

Need a reference mpi.h implementation

Need to ensure consistent calling and linkage conventions across binary implementations for a particular platform

Need to come to consensus on what level standard – part of MPI 3.0 standard as a whole, or an add-on (MPI ABI compliant)

Some technical detail to be addressed extracted from Bill Gropp's GMPI paper

http://www.cs.uiuc.edu/homes/wgropp/bib/papers/2002/gmpishort.pdf

- Compatible MPI data entities
  - Compile-time values
  - Init-time constants (MPI_INT, etc.)
  - Opaque objects (MPI_Comm, MPI_Datatype, etc.)
  - Defined objects (MPI_Status)
  - Defined pointers (MPI_BOTTOM, MPI_STATUS_NULL, etc.)
- Uniform approach to the macro implementation of certain functions
- Similar calling convention
  - Argument order and size
  - Stack frame management policy
  - Return address storage and handling
  - Function call and return handling
- Common linkage convention
  - Library file format
  - MPI library name
  - MPI library path resolution mechanism
  - System library dependency resolution

Next Steps

Develop a concrete proposal to be presented at a future (possibly the next) forum meeting

Current members of the ABI working group

alexander.supalov@intel.com
david.solt@hp.com
Dries.Kimpe@cs.kuleuven.be
edric.ellis@mathworks.co.uk
erezh@microsoft.com
gil@dev.mellanox.co.il
howardp@cray.com
jeffb@lanl.gov
jsquyres@cisco.com
kannan.narasimhan@hp.com
keller@hlrs.de
koziol@hdfgroup.org
lindahl@pbm.com
sameer@cs.uoregon.edu
terry.dontje@sun.com

To join us: http://lists.mpi-forum.org/mailman/listinfo.cgi/mpi3-abi “**Subscribing to mpi3-abi** “