Using MPICH for Fun and Profit

Jeff Hammond
Principal Architect
HPC Software

Outline

1. MPI ABI Collaboration
2. MPI Fortran 2008 (VAPAA)
3. ~~MPI-3 RMA (ARMCI-MPI)~~

MPI ABI

MPI ABI Standardization

Goal: interoperability between implementations: build once, run many.

History:

- 2006: users want a common or standard ABI
  - 2016: CEA wi4mpi project began
  - 2021: Erik Schnetter creates MPI Trampoline
  - 2021: ABI standardization effort begins
  - 2023: I created Mukautuva, Hui adds ABI prototype to MPICH

MPI Application Binary Interface Standardization

Jeff R. Hammond
NVIDIA Helsinki Oy
Helsinki, Finland
NVHPC SDK, Fortran

Lisandro Dalcin
Extreme Computing Research Center
KAUST
Thuwal, Saudi Arabia
dalc l.com
Python

Erik Schnetter
Perimeter Institute for Theoretical Physics
Waterloo, Ontario, Canada
esc e.ca
Julia, MPItrampoline

Marc Pérache
CEA, DAM, DIF
Arpajon, France
wi4mpi, containers, MPC

Jean-Baptiste Besnard
ParaTools
Bruyères-le-Châtel, France
jbbes s.fr
TAU, E4S

Jed Brown
University of Colorado Boulder
Boulder, Colorado, USA
j g
PETSc, Rust

Gonzalo Brito Gadeschi
NVIDIA GmbH
Munich, Germany
Rust, containers

Joseph Schuchart
University of Tennessee, Knoxville
Knoxville, Tennessee, USA
sc du
Open MPI

Simon Byrne
California Institute of Technology
Pasadena, California, USA
simonby tech.edu
Julia

Hui Zhou
Argonne National Laboratory
Lemont, Illinois, USA
zh ov
MPICH

Open Access Paper
https://dl.acm.org/doi/10.1145/3615318.3615319

Current Status

MPICH supports the proposed ABI, as defined in the reference header; tested with mpi4py, etc.

MPI Forum still debating fine details of Fortran support.

As a side effect of the ABI effort, MPICH test suite is implementation-agnostic and can be used to test Open MPI, e.g.

https://github.com/mpiwg-abi/header_and_stub_library/


NVIDIA
VAPAA

VAPAA

In Finnish, Vapaa means "free", in the sense of "free-range chickens."

What:

Standalone implementation of MPI Fortran support (**MPI_F08**).

Why:

Workaround Fortran compiler and MPI implementation issues to get all the features everywhere.

How:

Use MPI C API; translate subarrays to datatypes using CFI_cdesc_t.
Use MPICH's MPIX_Type_iov instead of tedious and slow type introspection with MPI API.

When:

Common features are available. Features added based on user interest. Code generation will achieve feature-completeness eventually.

https://github.com/jeffhammond/vapaa


NVIDIA