Objectives

- Introduce you the fundamentals of MPI by FORTRAN and C examples
- Provides overview of MPI basics
- Teaches you how to compile, link and run MPI codes
- Learning basic MPI functions
Outline

• Introduction to MPI
• Basics of MPI
• Compilation and Execution
• Learning by examples (in Fortran or C)
• Reference
Introduction to MPI

- Message passing interface (MPI)
  - Parallel computing utility library
  - Contains many subroutine/functions
  - Not a independent language
  - MPI subroutines/functions can be invoked from Fortran and C, respectively
  - With FORTRAN or C compilers
  - MPI-1 doesn’t support F90, but MPI-2 does support Fortran and C++
Introduction to MPI (cont.)

• Why people use MPI?
  – Speed up computation
  – Big demand of CPU time and more memory
  – More portable and scalable rather than using automatic "parallelizer"
  – Good for distributed computing systems, such as distributed clusters, network based computers (or workstations)
  – More or least "standard"
Disadvantages using MPI?

- More complicated than serial computing
- More complicated to master the technique, especially parallel algorithms
- Synchronization lost
- Amount of time required to convert a serial code to a parallel code
Introduction to MPI (cont.)

- Alternative ways?
  - Data parallel model using high level language such as HPF
  - Advanced library (or interface), such as (The Portable, Extensible Toolkit for Scientific Computation (PETSC)
  - Java multithread computing on internet based distributed computation, such as Common Object Request Broker Architecture (CORBA) and Grid computing
Basics of MPI

- MPI header (library) file should be included in user’s FORTRAN or C codes.
- The library files contain definitions of constants and prototypes.

```
#include "mpif.h" for FORTRAN code
#include "mpi.h" for C code
```
Basics of MPI

• MPI is free
• Where you can download or get MPI?
  – MPI standards (http://www.mcs.anl.gov/research/projects/mpi)
  – LAM.MPI Parallel Computing (http://www.lam-mpi.org/)
Basics of MPI

- Mississippi State University NSF Engineering Research Center (http://www.hpc.msstate.edu/)
- MPI-lite (UCLA) (http://pcl.cs.ucla.edu/projects/sesame/mpi_lite/mpi_lite.html)
- Wiki about MPI (http://en.wikipedia.org/wiki/Message_Passing_Interface)
Basics of MPI

- UTK (MPI page)
  (http://www.cs.utk.edu/~tatebe/research/mpi/)

- Open MPI (http://www.open-mpi.org/)

- MPI Complete Reference at Netlib

- IINL’s MPI
  (https://computing.llnl.gov/tutorials/mpi/)
Basics of MPI

- MPICH2
  (http://www.mcs.anl.gov/research/projects/mpi
  ch2/)
- Open MPI (http://www.open-mpi.org/)
- MP-MPICH for heterogeneous clusters
  (http://www.lfbs.rwth-aachen.de/content/mp-
  mpich)
- WMPI-II (http://www.criticalhpc.com/)
Basics of MPI (cont.)

• Basic idea behind MPI is to send (or receive) messages from one process to another process
Basics of MPI (cont.)

- MPI segments can be included inside the serious code to make the execution of the code in parallel.
- The structure of MPI enhanced code
Basics of MPI (cont.)

- MPI is initiated by calling `MPI_Init()` first before invoking any other MPI subroutines or functions.
- MPI processing ends with a call `MPI_Finalize()`.
Basics of MPI (cont.)

• Only difference between MPI subroutines (for FORTRAN) and MPI functions (for C) is the error reporting flag.
  – In FORTRAN, it is returned as the last member of the subroutine's argument list.
  – In C, the integer error flag is returned through the function return value.
  – Consequently, MPI FORTRAN subroutines always contain one additional variable in the argument list than the C counterpart.
Basics of MPI (cont.)

- C's MPI function names start with MPI_ followed by a character string with the leading character in upper case letter while the rest in lower case letters.
- FORTRAN subroutines bear the same names but are case-insensitive.
- On SGI's Origin20000 (NCSA), parallel I/O is supported.
Compilation and Execution (f77)

- To compile and execute a f77 (or f90) code without MPI

  f77 -o example example.f
  /bin/time example

- To compile and execute a f77 (or f90) code with MPI

  f77 -o example1_1 example1_1.f -lmpi
  /bin/time mpirun -np 4 example1_1
Compilation and Execution (C)

- To compile and execute a C code without MPI
  
  \[
  \text{cc -o exampleC exampleC.c -lm}  
  \]
  
  
  \[
  \text{/bin/time exampleC}  
  \]

- To compile and execute a C code with MPI

  \[
  \text{cc -o exampleC1_1 exampleC1_1.c -lmpi -lm}  
  \]

  \[
  \text{/bin/time mpirun -np 4 exampleC1_1}  
  \]