Fortran with MPI

What is it? How to get started, and details

Arun Prasaad Gunasekaran

Image Sources:https://computing.llnl.gov/tutorials/parallel_comp/

Outline

- What to Expect?
- What is MPI?
- How does it work?
- Supported Languages?
- Requirements
- Installation
- Pros/Cons
- Alternatives

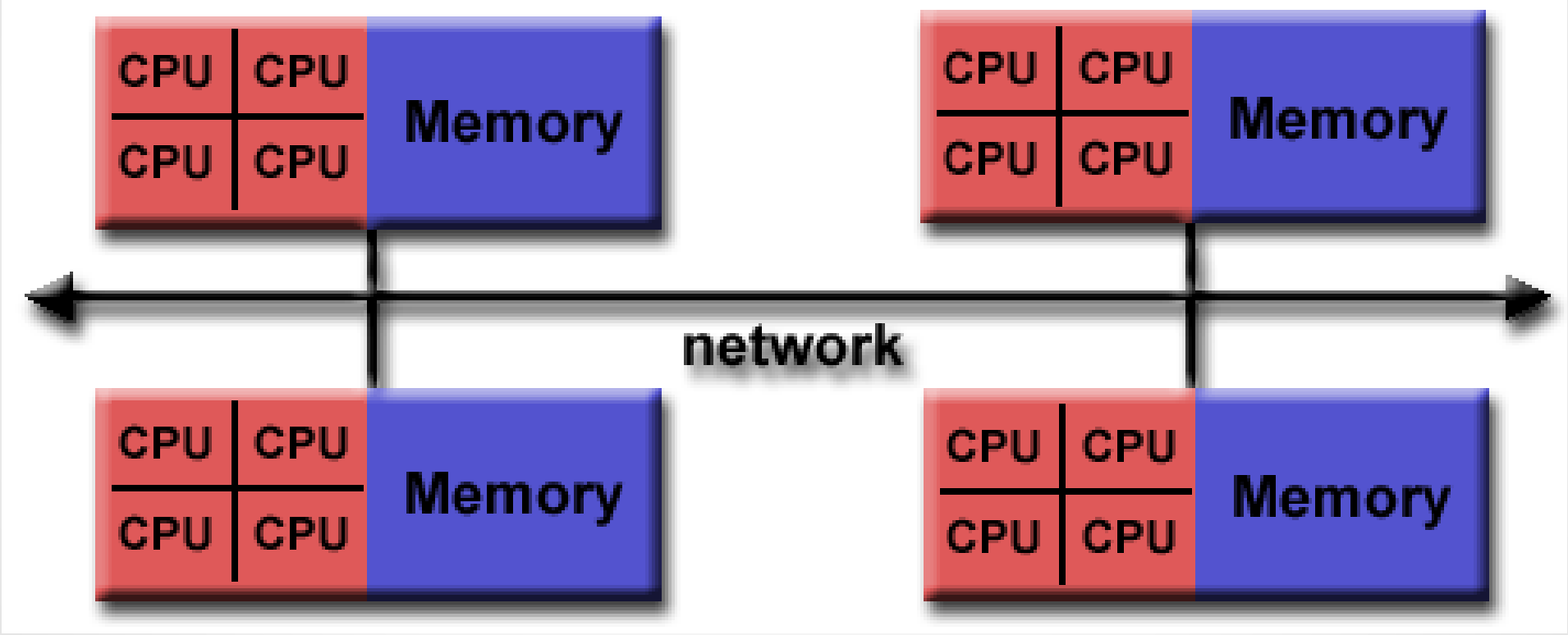
What to Expect?

- Covers some vital commands
- Semi-Realistic examples
- Designed to give a good motivation into the topic
- Covers basics with a few advanced topics
- For Beginners
  - Motivating Examples
  - Explainer
  - Reference
- For Advanced Users
  - Reference
  - Brush-up

What is MPI?

- MPI – Message Passing Interface.
- It works on top of a compiled programming language.
- Created for enabling Parallel Programming.
- Works on Distributed Memory Systems.
- The program runs in all processors.
- Parallelized programs are much different from serial programs.
- Selective portions are done by different processors
- Results and data are communicated (“passed”) between processes as packets of data (“messages”) through the network (“Interface”)
- Hence the name “Message Passing Interface”

How does it work? - Distributed Memory Systems



How does it work?

- Each process runs the program separately,
- Individual processes communicate with other processes requesting data,

Supported Languages

- Compiled Languages - C, C++, Fortran
- Also available in Python (MPI4PY)

Requirements

- Compilers
  - GNU (gcc, g++, gfortran)
  - Intel (icc, icpc, ifort)
- Atleast One MPI Implementation
  - OpenMPI
  - MPICH
  - MVAPICH

Installation

Will be using GNU Fortran with MPICH

- Installting gnufortran
- Installing mpich

Verification

- Check for gfortran, mpirun, mpiexec, mpicc, mpi++, mpif90

Pros/Cons

- Scalable
- Portable
- Avoids Race Condition
- Can run in both Shared and Distributed Memory Systems
- Cannot compile the code serially
- Speed limited by communication lags
- Tendency to become complicated

Alternatives

- OpenMP
- OpenACC
- GPU computing