

Introduction to MPI Programming

Kameswararao Anupindi

Department of Mechanical Engineering
Indian Institute of Technology Madras (IITM)

July 27, 2020



Outline

- Introduction
- Basic functions/routines
- Example programs
- Further learning/resources
- Closure

Introduction: MPI - Message Passing Interface

- MPI is a library that is the de facto standard for distributed memory computing
- MPI consists of a collection of routines
- These routines provide exchange of data and synchronization among the processes
- MPI functions are highly optimized
- Most supercomputers run MPI (together with other parallelisms)

Top supercomputers in the world - top500.org



(a) 1. Fugaku



(b) 2. Summit

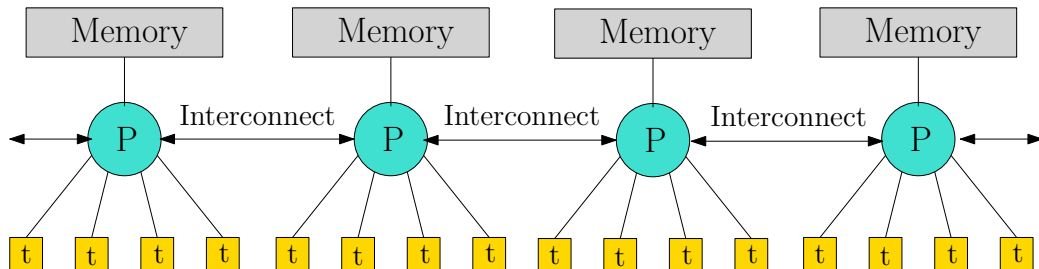


(c) 3. Sierra

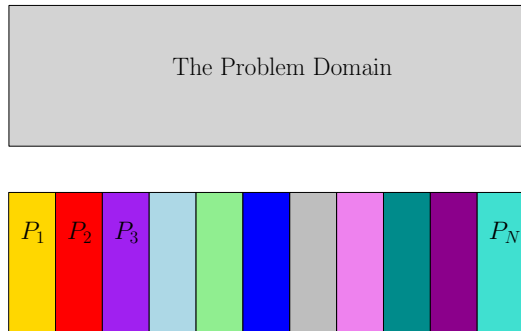


(d) 66. Pratyush

The parallel configuration



Problems that yield to parallelization



- Problems that can be sliced up
- SPMD - single program multiple data
- Data exchange between processors
- Ponder on problems that cannot be parallelized?

Four basic functions in C

```
MPI_Init(&argc, &argv);
```

```
MPI_Comm_size(MPI_COMM_WORLD, &size);
```

```
MPI_Comm_rank(MPI_COMM_WORLD, &myid);
```

```
MPI_Finalize();
```

Four basic functions in FORTRAN

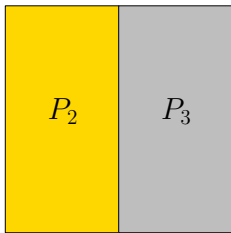
```
call MPI_INIT(mpierror)
```

```
call MPI_COMM_SIZE(MPI_COMM_WORLD, size, mpierror)
```

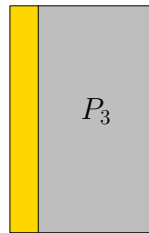
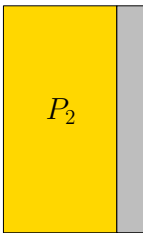
```
call MPI_COMM_RANK(MPI_COMM_WORLD, myid, mpierror)
```

```
call MPI_FINALIZE(mpierror)
```


A closer look at processor boundaries



(a) A pair of processes



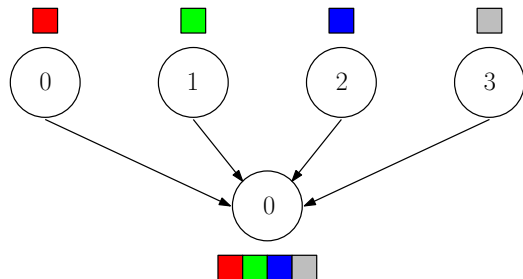
(b) Ghost layers

Functions for data exchange

```
MPI_Send(  
    void *data,  
    int count,  
    MPI_Datatype datatype,  
    int destination_rank,  
    int tag,  
    MPI_Comm communicator)
```

```
MPI_Recv(  
    void *data,  
    int count,  
    MPI_Datatype datatype,  
    int source_rank,  
    int tag,  
    MPI_Comm communicator,  
    MPI_Status *status)
```

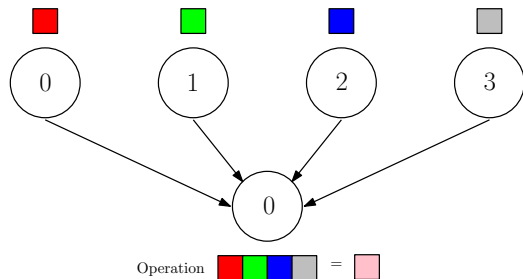
Collective communication: Gather



`MPI_Gather(`

```
void *send_data,  
int send_count,  
MPI_Datatype datatype,  
void *receive_data,  
int receive_count,  
MPI_Datatype datatype,  
int root,  
MPI_Comm communicator)
```

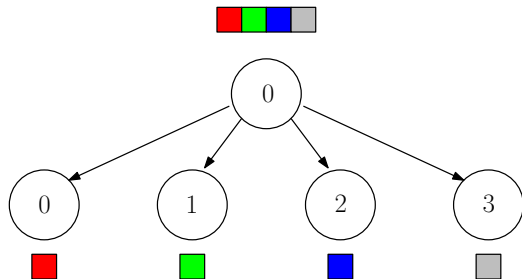
Collective communication: Reduce



`MPI_Reduce(`

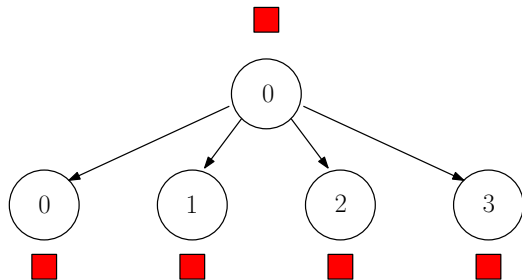
```
void *send_buffer,  
void *receive_buffer,  
int count,  
MPI_Datatype datatype,  
MPI_Op operator,  
int root,  
MPI_Comm communicator)
```

Collective communication: Scatter



```
MPI_Scatter(  
    void *send_data,  
    int send_count,  
    MPI_Datatype datatype,  
    void *receive_data,  
    int receive_count,  
    MPI_Datatype datatype,  
    int root,  
    MPI_Comm communicator)
```

Collective communication: Broadcast



`MPI_Bcast (`

```
void *data,  
int count,  
MPI_Datatype datatype,  
int root,  
MPI_Comm communicator)
```

- 1 Hello world in both C and FORTRAN
- 2 Example programs for gather, scatter, broadcast and reduce in C
- 3 Dot product of two vectors in both C and FORTRAN
- 4 One-dimensional Poisson equation using Jacobi iteration in FORTRAN

One-dimensional Poisson equation using Jacobi iteration

Governing equation:

$$\frac{d^2 u}{dx^2} = f(x) \quad (1)$$

$$f(x) = 2x \quad (2)$$

Boundary conditions:

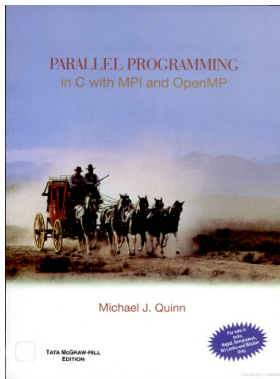
$$u(x = 0) = 1.0 \quad (3)$$

$$u(x = L) = 2.0 \quad (4)$$

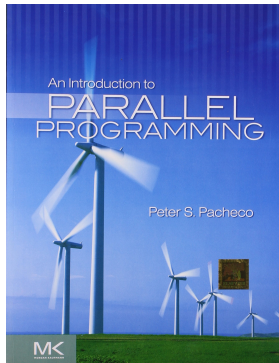
Jacobi iteration:

$$u_i^{k+1} = \frac{1}{2} \left(u_{i+1}^k + u_{i-1}^k - f(x_i)(\Delta x)^2 \right) \quad (5)$$

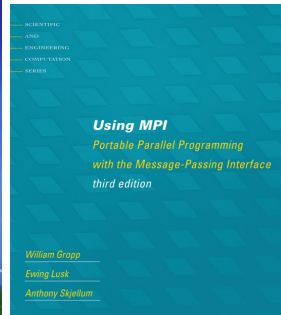
Further learning/resources - books



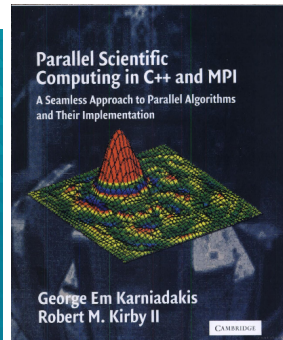
(a) Michael Quinn



(b) Peter Pacheco



(c) Gropp et al.



(d) Karniadakis & Kirby

Further learning/resources - online

- Cornell Virtual Workshop - <https://cvw.cac.cornell.edu/MPI/default>
- Lawrence Livermore National Lab MPI Tutorials
<https://computing.llnl.gov/tutorials/mpi/>
- MPI tutorial by William Gropp -
<https://www.mcs.anl.gov/research/projects/mpi/tutorial/gropp/talk.html>
- Upcoming Swayam course “High Performance Computing for Scientists and Engineers” by Prof. Somnath Roy - IITKGP - https://swayam.gov.in/nd1_noc20_me61/preview

- Basics of MPI programming using C and FORTRAN
- Ran example programs
- Sufficient to get started with “parallelizing your problem”
- Looked at resources for further learning

Thank You!
&
Questions?