

Programming with MPI

Mohammad Towhidul Islam
towhid@cs.umanitoba.ca

What is MPI?

- Message Passing Interface (MPI) is a library specification
- Set of functions for cooperation of parallel processes
- Freely available implementations include LAM and MPICH
- MPI is not a programming language

Simple Program

Coding in C:

```
#include <stdio.h>
int main (int argc, char **argv)
{
    printf("Hello");
}
```

Coding in C with MPI:

```
#include <stdio.h>
#include "mpi.h"
int main (int argc, char **argv)
{
    MPI_Init (&argc,&argv);
    printf("Hello");
    MPI_Finalize();
}
```

Output:

Hello
Hello

Setup an Account

- An account on Gull cluster
- Require three files
 - .cshrc (for tcsh or csh shell, it already exists)
 - .rhosts (need to create)
 - lamhosts (need to create)

Setup an Account (Cont.)

- At the end of `.cshrc` file, the following lines will be added:

```
setenv LAMHOME /usr/local/mpi
setenv PATH "$LAMHOME/bin:$PATH"
setenv MANPATH "$LAMHOME/man:$MANPATH"
setenv LAMRSH rsh
```

Setup an Account (Cont.)

.rhosts file

Format:

```
gull.cs.umanitoba.ca username
hawk.cs.umanitoba.ca username
loon.cs.umanitoba.ca username
owl.cs.umanitoba.ca username
robin.cs.umanitoba.ca username
sparrow.cs.umanitoba.ca username
swan.cs.umanitoba.ca username
falcon.cs.umanitoba.ca username
goose.cs.umanitoba.ca username
eagle.cs.umanitoba.ca username
```

Example:

```
gull.cs.umanitoba.ca towhid
hawk.cs.umanitoba.ca towhid
loon.cs.umanitoba.ca towhid
owl.cs.umanitoba.ca towhid
robin.cs.umanitoba.ca towhid
sparrow.cs.umanitoba.ca towhid
swan.cs.umanitoba.ca towhid
falcon.cs.umanitoba.ca towhid
goose.cs.umanitoba.ca towhid
eagle.cs.umanitoba.ca towhid
```

Permission : The permission of the file must be 644 (`rw-r--r--`)

Setup an Account (Cont.)

- `lamhosts` file (The name of the file can be anything)

```
gull
hawk
loon
owl
robin
sparrow
swan
falcon
goose
eagle
```

Permission : The permission of the file must be 644 (`rw-r--r--`)

Boot the Machines !

- Purpose:**
 - LAM uses a daemon on each node for process control, message passing. "Booting LAM" means the launch of these daemons in each node.
- Command:**
 - `lamboot -v lamhosts`
- Output:**

```
LAM 6.5.8/MPI2 C++/ROMIO - Indiana University
Executing hboot on n0 (gull - 1 CPU)...
Executing hboot on n1 (hawk - 1 CPU)...
Executing hboot on n2 (loon - 1 CPU)...
Executing hboot on n3 (owl - 1 CPU)...
Executing hboot on n4 (robin - 1 CPU)...
Executing hboot on n5 (sparrow - 1 CPU)...
Executing hboot on n6 (swan - 1 CPU)...
Executing hboot on n7 (falcon - 1 CPU)...
Executing hboot on n8 (goose - 1 CPU)...
Executing hboot on n9 (eagle - 1 CPU)...
topology done
```

Compile & Run

- Compile:
 - mpicc -o test.o test.c
- Run:
 - mpirun -np <number of processes> <program name>
 - mpirun -np 2 test.o
 - Another variation:
 - mpirun n0-3 -c test.o

More than Basic

```
#include <stdio.h>
#include "mpi.h"
void main( int argc, char **argv )
{
    int rank; /* Rank of process*/
    int size; /*Number of processes*/
    MPI_Init( &argc, &argv );
    MPI_Comm_size( MPI_COMM_WORLD, &size );
    MPI_Comm_rank( MPI_COMM_WORLD, &rank );
    printf( "Hello world from process %d of %d\n",rank, size );
    MPI_Finalize();
}
```

Output

```
Hello world from process 0 of 4
Hello world from process 1 of 4
Hello world from process 2 of 4
Hello world from process 3 of 4
```

Thanks
Questions?