Parallel Architectures and Systems
MPI Information at UTD, 2010

Dr. Edwin Sha

If you have any question about MPI, first read http://www.lam-mpi.org/tutorials/. Or go to openMPI web site to find FAQ.

We have no problem in running MPI in our parax Linux machines. If you have any problem, please ask TA.

1 Setting up the Environment for open MPI

We will be using the openMPI public domain implementation of MPI. It is most commonly used on clusters of workstations, such as the SUN machines that we have here in the department.

The machines we are using are departmental Linux machines from **para1** to **para8**. You should "ssh -l username" to remotely login to one of these machines

In your $HOME directory, create a file called *.rhosts* that contains the following lines.

```
para1.utdallas.edu
para2.utdallas.edu
para3.utdallas.edu
para4.utdallas.edu
para5.utdallas.edu
para6.utdallas.edu
para7.utdallas.edu
para8.utdallas.edu
```

Do you know what is the purpose of this .rhosts file?

Next, add in your `.bashrc` file the following:

```
export PATH=$PATH:/usr/lib64/openmpi/1.2.7-gcc/bin
```

Make sure they are above the following line of `.bashrc`:

```
if [ -z "$PS1" ] ; then
```

You also need to create an SSH identity to run MPI without typing a password or a passphrase too many times.

```
$ ssh-keygen -t rsa
Enter file in which to save the key ($HOME/.ssh/id_rsa):  enter to use default.
Enter passphrase (empty for no passphrase):  enter for no passphrase.
Enter same passphrase again:  enter again.

$ cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

Then, you need to make sure that the permission of your home directory is set appropriately. Login to para1.utdallas.edu and run the following commands in your home directory.

```
$ ll .. | grep xxx000000 (replace with your own netid)
```

The result should be something like this:
drwx–x–x 47 xxx000000 **cs** 2560 Mar 1 16:32 xxx000000

Pay attention to the group ownership. We assume it is cs here.

If the permission bits are not right, run the following command:

```
$ chmod 711 ../xxx000000
```

If the group ownership is not **cs or ce or ee**, run the following command (You need to replace cs with ce or ee, if you are not a cs student):

```
$ chgrp -R cs ../xxx000000
```

Use "cs" or "ce" or "ee" according to which academic program you are in.

Now, the first time you run `mpirun`, you may need to save the RSA key fingerprints of all the hosts specified in `hostfile`. At every `Are you sure you want to continue connecting (yes/no)?`, enter `yes`. I would suggest you first "ssh -l usename para1." Then in para1, you manually ssh to para2, para3, ... to para8. So the RSA key can be saved.

2 Writing Your First Program

You will also find a sample MPI program, `dpi.c`, in the directory (`/net/core/export/home/cs/001/e/edsha/parallel/openMPI`). It demonstrates simple MPI message passaging features. Please feel free to copy all the files there to your local directory. You can use the following UNIX command to copy all the files into your "mpi" local directory.

```
cp -rf ~edsha/parallel/openMPI mpi
```

2.1 Calculating PI Using Numerical Integration

This program approximate the value of π by numerical integration[1] on

$$\int_0^1 \frac{1}{1+x^2} dx = \frac{\pi}{4}$$

```
#include <math.h>
#include "mpi.h"

double f(double a)
```

[1] Gropp, W., Lusk, E., and Skjellum, A. *Using MPI.* (Cambridge, MA: MIT Press, 1995).

```
{
  return (4.0 / (1.0 + a*a));
}

int main(int argc, char *argv[])
{
  int done = 0,  n,  myrank,  numprocs,  i,  rc;
  double PI25DT = 3.14159265358979323846264 3;
  double mypi,  pi, h, sum, x, a;
  double startwtime, endwtime;

  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
  MPI_Comm_rank(MPI_COMM_WORLD, &myrank);

  n = 0;
  while (!done) {
    if (myrank == 0) {
      if (n == 0)
        n = 100000;
      else
        n = 0;
      startwtime = MPI_Wtime();
    }
    MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
    if (n == 0)
      done = 1;
    else {
      h   = 1.0 / (double) n;
      sum = 0.0;
      for (i = myrank + 1; i <= n; i += numprocs) {
        x = h * ((double)i - 0.5);
        sum += f(x);
        dummy(n) // I put a dummy function to increase computation load
      }
      mypi = h * sum;

      MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);

      if (myrank == 0) {
        printf("pi is approximately %.16f, Error is %.16f\n",
               pi, fabs(pi - PI25DT));
        endwtime = MPI_Wtime();
```

```
        printf("wall clock time = %f\n",
               endwtime-startwtime);
      }
    }
  }
  MPI_Finalize();
  return 0;
}
```

- The command `MPI_Init(&argc, &argv)` initializes the MPI task. This command is necessary for all MPI programs.
- `MPI_Comm_size(MPI_COMM_WORLD, &numprocs)` is used to obtain the number of processors `mpirun` associated with the `MPI_COMM_WORLD` communicator.
- `MPI_Comm_rank(MPI_COMM_WORLD, &myrank)` is analogous to the PVM `pvm_mytid()` – it obtains the id of the current task within the `MPI_COMM_WORLD` communicator.
- `MPI_Wtime()` returns the wall clock time.
- `MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD)` performs a broadcast (similar to `pvm_bcast()`) to/from all the ranks in the `MPI_COMM_WORLD` communicator.
- `MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD)` Reduces values on all ranks to a single value – the reduction operation shown here is a global sum.
- `MPI_Finalize()` gracefully shuts down the MPI process. This command is necessary for all MPI programs. You still need to `exit()` or return from `main()`.

3 Compiling and Running Your MPI Program

You should "telnet" ("ssh -l usename") to a paraX machine where X can be 1 to 8 to compile and run your program. Make sure you "ssh" to one of the machines listed in your hostfile that is described later.

For how to compile and run your MPI programs, check http://www.open-mpi.org/faq/

3.1 `mpicc`

You also can simply use the following command to compile your program.

```
mpicc -o dpi dpi.c
```

3.2 The Hostfile

To run your MPI program, you first need to create a hostfile. Check the MPI tutorial to get more details about how to set up this file. Here is a sample hostfile (hosts):

```
para1.utdallas.edu
para2.utdallas.edu
para3.utdallas.edu
para4.utdallas.edu
```

The machine that you will be launching the program from should be listed first in the file.

3.3 Running Your MPI Program

First read the README file. And **ssh -l username** to a paraX machine to run your program.

It is very easy to run your program.

```
mpirun --hostfile hosts -np 2 dpi
```

Try different -np number, say "-np 1", "-np 4" to see the different performance.

Try several times if ssh prompt (yes/no) appears. If it hangs, don't worry. Just CTRL-C and repeat mpirun again.