MATLAB Laboratory for MPI toolbox (MPITB)

Morris Law

November 6, 2003

1 Introduction

MPITB – MPI toolbox for MATLAB was written by Javier Baldomero, when he was doing his PhD research in University of Granada, Spain. With this toolbox, parallelization of Matlab codes can be done with Message Passing Interface (MPI) standard. MPI toolbox can be run in Network of workstations and PC cluster running Linux or Solaris. In our University, the Scientific Computing Laboratory and the PC cluster in Science Faculty has installed the MPITB. In this laboratory, the workstations in T713 and PC cluster will be used as a common interface for teaching and demonstrating the usage of this toolbox. An example of estimating π using Monte-Carlo simulation is introduced.

The toolbox came with precompiled binary on Linux. It worked with MATLAB 6.1 and LAM 6.5.4. To run MATLAB 6.1 in the Scientific Computing Laboratory, user must type `matlabr12`. To run MATLAB 6.1 in our PC cluster, user can use the command `matlab`. Therefore, the work examples in this laboratory session have been prepared in two files, **MC-scilab.tar.gz** for workstations in Scientific Computing Laboratory and **MC-cluster.tar.gz** for the PC cluster. These files can be downloadable from the subject homepage (http://www.math.hkbu.edu.hk/math2160/).

2 Preparations

In order to run MPITB on MATLAB, one must use LAM/MPI — the Local Area Multicomputer software to set up topology so that a number of computers are used to solve your problem at one time.

2.1 Login to cluster

Student accounts have been opened in the P4-Xeon cluster for the demonstration. A number of 3-4 students may form a group. Login the cluster using security shell from one of the workstations in SCT713. For instance,

```
    ssh -l 02001234 tdgrocks
```

2.2 Download and untar the work examples

For workstations in the Scientific Computing Laboratory T713, in the subject homepage, download the file `MC-scilab.tar.gz` and place it in your home directory. Uncompress and untar it with the following command,

```
        tar xzvf MC-scilab.tar.gz
```

the directory matlab/MC will be created and stored the work examples. In additions, the hostfile named `lambhost` and a file named `bashrc.mpitb` will appear in the home directory.

For the PC cluster, the above files has already been prepared and placed in the student accounts assigned to you. You may also download the file `MC-cluster.tar.gz` and untar as above.

2.3 Hostfile: lambhost

Prepare a LAM hostfile called `lambhost` in your home directory using an editor to store the following hostnames. For T713 workstations, the file is similar as below,

```
spc25.sci.hkbu.edu.hk
spc26.sci.hkbu.edu.hk
spc27.sci.hkbu.edu.hk
spc28.sci.hkbu.edu.hk
spc29.sci.hkbu.edu.hk
spc30.sci.hkbu.edu.hk
```

Use an editor to insert and modify the content of the hostfile to suit your need. Usually, user should place the login workstations to the bottom of the list. Besides, the total number of process can be run is restricted to the number of CPU you assigned. You can spawn a number of 6 MATLAB process at one time with the above hostfile.

For the P4-Xeoncluster, the files show

```
comp-pvfs-01 cpu=2
comp-pvfs-02 cpu=2
comp-pvfs-03 cpu=2
comp-pvfs-04 cpu=2
comp-pvfs-05 cpu=2
frontend-0
```

You may add more hostnames in it. It is preferred to place the name frontend-0 at last.

2.4 Procedure for ssh-keygen

In order to use multiple computers, the steps to generate ssh key pair shall be done in the workstations. For cluster use, the keypair will be setup after you first login.

1. ssh-keygen -t rsa
2. chmod 755 .ssh
3. cd .ssh
4. cp id_rsa.pub authorized_keys2
5. chmod 644 authorized_keys2

2.5 Boot up the parallel architecture using LAMBOOT

Use the command `lamboot -v` to boot up the LAM topology.

```
lamboot -v
```

Please make sure that the words *topology done* was reported after you run the above command. You may double check using 'ps -ef | grep lamd' to see your process running.

Contragulation, you have prepared a parallel architecture for computing. Once you finished, remember to clean up the topology by lamhalt, lamclean or wipe.

2.6 Environment variables for MPITB

The following environment variable are essential for MPITB. Please check if the following are defined by env | grep *variable*.

MPITB_ROOT	/usr/local/matlab/toolbox/mpitb
LAMRSH	ssh -x
LAMBHOST	~/lambhost

If they have not been defined, please insert the following lines in your shell startup file `.bashrc` in your home directory. The file `bashrc.mpitb` is for your reference.

```
MPITB_ROOT = /usr/local/matlab/toolbox/mpitb
LAMRSH = 'ssh -x'
LAMBHOST = ~/lambhost
export MPITB_ROOT LAMRSH LAMBHOST
```

Once you modified your `.bashrc` file, please logout and login again.

2.7 Sample files for MPITB toolbox

Several sample files have been copied in each user's account. Locate them in matlab/MC under your home directory.

MCpi.m	:	The monte-carlo PI simulation programme in serial
MCpiMast.m	:	The master program for estimating PI in parallel
MCpiWork.m	:	The slave program for estimating PI in parallel
startup.m	:	The startup program used by MATLAB in setting up parameters for MPITB
startup_piSlv.m	:	The startup program used by MATLAB in running child processes in slave nodes
Mast.m, Work.m	:	sample programs come with MPITB for calculating π by integral of $\frac{1}{1+x^2}$

3 Monte Carlo estimation of PI

3.1 Mathematical background

$$\int_{[0,1]^2} f(x)dx = \text{Volume of the region under the surface z = f(x) on top and z = 0on the bottom } f(x) \text{ for } x \in [0,$$

The Monte Carlo method can be explained as follow
Choose $x_1, \cdots, x_n$ iid $u\left([0,1]^2\right)$

$$I = \int_{[0,1]^2} f(x)dx \simeq \frac{1}{n}\sum_{i=1}^{n} f(x_i) = \hat{I}$$

$$E(\hat{I}) = E\left[\frac{1}{n}\sum_{i=1}^{n} f(x_i)\right] = \frac{1}{n}\sum_{i=1}^{n}\int_{[0,1]^2} f(x)dx = I$$

Therefore, $\hat{I}$ is an unbiased estimator of I.

$$\begin{aligned} \text{var}(\hat{I}) &= E\left[(I-\hat{I})^2\right] = E\left[I^2 - 2I\hat{I} + \hat{I}^2\right] \\ &= I^2 - 2I^2 + E(\hat{I}^2) = E(\hat{I}^2) - I^2 \end{aligned}$$

$$
\begin{aligned}
E(\hat{I}^2) &= E\left[\frac{1}{n}\sum_{i=1}^{n} f(x_i)\frac{1}{n}\sum_{j=1}^{n} f(x_j)\right] \\
&= E\left[\frac{1}{n^2}\sum_{i,j=1}^{n} f(x_i)f(x_j)\right] \\
&= \frac{1}{n^2}\sum_{i,j=1}^{n} E\left(f(x_i)f(x_j)\right)
\end{aligned}
$$

$$
E\left(f(x_i)f(x_j)\right) = \begin{cases} E\left\{[f(x_i)]^2\right\} = \int_{[0,1]^2} f(x)^2 dx & i = j \\ E\left[f(x_i)\right] E\left[f(x_j)\right] = \left[\int_{[0,1]^2} f(x) dx\right]^2 & i \neq j \end{cases}
$$

$$
\begin{aligned}
E(\hat{I}^2) &= \frac{1}{n^2}\left[n\int_{[0,1]^2}[f(x)]^2\,dx + (n^2-n)\left[\int_{[0,1]^2} f(x)dx\right]^2\right] \\
&= \frac{1}{n}\int_{[0,1]^2}[f(x)]^2\,dx + \left(1-\frac{1}{n}\right)I^2
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{var}(\hat{I}) &= \frac{1}{n}\int_{[0,1]^2}[f(x)]^2\,dx - \frac{1}{n}I^2 \\
&= \frac{1}{n}(\int_{[0,1]^2}[f(x)]^2\,dx - I^2) = \frac{1}{n}\mathrm{var}(f)
\end{aligned}
$$

Root mean square error (RMSE) $= \sqrt{\mathrm{var}(f)}/\sqrt{n}$

For the special case

$$
f(x) = \begin{cases} C & x \in A \\ 0 & x \notin A \end{cases} \Rightarrow \int_{[0,1]^2} f(x)dx = C = \mathrm{Area}(A)
$$

$$
\begin{aligned}
[f(x)]^2 &= f(x) \\
\mathrm{var}(f) &= I - I^2 \\
&= I(C - I)
\end{aligned}
$$

Consider the case of computing π using the following

$$
\begin{aligned}
f(x) &= \begin{cases} 4 & x^2+y^2 \leq 1 \\ 0 & x^2+y^2 > 1 \end{cases} \\
I &= \int_{[0,1]^2} f(x)dx = \pi
\end{aligned}
$$

$$\begin{aligned} f(x) &= \begin{cases} 4 & x^2+y^2 \le 1 \\ 0 & x^2+y^2 > 1 \end{cases} \\ [f(x)]^2 &= 4f(x) \\ \mathrm{var}(f) &= 4I - I^2 = I(4-I) \\ \mathrm{var}(\hat{I}) &= I(4-I)/\sqrt{n} \end{aligned}$$

3.2 Programming

π can be estimated by the area of circle with unit radius. Assuming that within a unit square, a number of random points are chosen. The more the no. of random points are chosen, the ratio of no. of points inside the quadrant and the total no. of points can be a good estimate of $\pi/4$.

The program MCpi.m was written to estimate π in serial as follow,

```
g=inline(’x.^2+y.^2’,’x’ ,’y’);
tic
n=1000000; procrank=1;
rand(’state’,sum(100*clock)+procrank*n)
x=rand(n,1); y=rand(n,1);
piapp=(4/n)*sum(g(x,y)<=1,1)
stderrest=sqrt(mean(piapp*(4-piapp)/n))
%stderr=sqrt(pi*(4-pi)/n)
toc
```

The corresponding program in parallel has been written with two separate Matlab function named MCpiMast.m for master program and MCpiWork.m for the slave program. User can called MCpiMast(C,N) where C is the number of slave node to run MCpiWork.m and N is the number of random points for estimate. Normally N shall be large to attain a more accurate estimate. If C=0, MCpiMast will run in serial. That is, running MCpiMast(0,10000) will run an estimate of PI in serial with 10000 random chosen points.

4 Running π estimation in serial

Run the sample program MCpi.m in both T713 workstations and the cluster. Adjust the number of random samples used and find out the π and the error estimates. Can you see the difference with different number of samples?

5 Running π estimation in parallel

Make sure you have lamboot using the hostfile, change directory to matlab/MC, check the directory to see if the sample files are there, then invoke matlab release 6.1 in that directory by

```
    matlabr12            % in T713 workstations (e.g. spc25)
    matlab               % in PC cluster frontend
```

The following message will appear in the MATLAB command windows,

```
MPI default hostfile: /users/staff/dean/morris/lamhosts
Set options to Homogeneous+Client-to-Client with the command:
  putenv(['TROLLIUSRTF=' int2str(RTF_HOMOG+RTF_MPIC2C)]), MPI_Init
Help on MPI: help mpi MCpi MCpiMast MCpiWork Mast Work
>>
```

5.1 Run MCpiMast(C,N)

The MATLAB program MCpiMast.m was written to work with MCpiWork.m to estimate π using Monte-Carlo Simulation. When run, the master program `MCpiMast(C,N)` will use `C` computers (nodes) and `N` random samples. In each of the `C` computers, the slave process (worker) will use a number of $n = N/C$ random samples to estimate π partially and send back to the master program. The aggregate estimate will be calculated and the result of estimation will be printed out with error estimate and time for calculation.
Run MCpiMast(C,N) for a number of computers and different N. See if there is any speedup. Try to compare the performance of the workstations in T713 and the PC cluster.

5.2 Efficiency in parallel programming

Let T be the time used for serial jobs and t be the time used to calculate in parallel in C computers. The efficiency of the parallel computer is defined by

$$\frac{T}{t \times C}$$

The performance is high as the efficiency approaches 1.

The MATLAB program MCeff.m has written to calculate the efficiency of the parallel architecture. Run MCeff to get a efficiency graph of 0 - nproc computers. Modify the content of lambhost to include enough computer for varying nproc.

6 Exercise

1. Run MCeff.m with different values of $N = 10^3, 10^4, 10^5, 10^6$ and 10^7 for nproc = 12. Find out the smallest value of N that efficiency > 70%?

2. Given the following function $f(x, y)$, $0 < x, y < 1$,

$$f(x,y) = 5e^{-2((x-\frac{1}{3})^2+(y-\frac{2}{3})^2)} - 3e^{-((x-\frac{3}{4})^2+(y-\frac{1}{4})^2)}$$

   By modifying MCarMast.m and MCarWork.m, find out the Monte Carlo estimation of $f(x, y) \geq 1$. Run MCarMast(5,1e7) and record the result.

7 Submit your result

Prepare a directory call matlab5 in your home directory in T713, store all your files in it and type submit matlab5 to hand in your homework before November 21, 2003.

8 For fun, run Mast(C,N,'s','d')

The MATLAB programs Mast.m and Work.m came with MPITB toolbox by Javier. It is yet another method of π estimate that is suitable for parallel programming practice. This method is to estimate π with the integral $4\int_{x=0}^{1} 1/1 + x^2dx$. The master program will spawn out worker programs and run on C computers. Given N divisions of strip under the function curve, each computers will sum up the area N/C strips and then send back the partial results to the master program. Then the aggregate area will be calculated as an estimation of π. Try the following for your interest,

```
Mast(0,100000,'s','d')
Mast(2,100000,'s','d')
Mast(4,100000,'s','d')
Mast(8,100000,'s','d')
```

to see if there is any speed up with more computers used.

9 Remember to LAMCLEAN the parallel architecture and then logout

Remember that the clean up the parallel architecture when you finished programming and quit MATLAB. Type lamhalt, lamclean or wipe can clean up the parallel architecture.

```
lamhalt -v
lamclean
wipe -v
```

The above commands are the same.

10 Reference

- MPITB homepage (http://atc.ugr.es/javier-bin/mpitb_eng)
- LAM/MPI parallel program (http://www.lam-mpi.org)
- MPITB tutorial written by Tammy Lam (http://www.sci.hkbu.edu.hk/∼smlam/MPITB/
- Tutorial given by MPITB toolbox $MPITB_ROOT/TUTORIAL