Programming Assignment III: MPI Programming

The purpose of this assignment is to familiarize yourself with MPI programming.

1 Statement of Problem 1

In this problem, you need to use MPI to parallelize the following serial program (`http://www.cs.nctu.edu.tw/~ypyou/courses/PP-f17/assignments/HW3/prime.c`), which takes a `long long int` argument as an input, finds the largest prime number that is smaller than the input, and counts the prime numbers that are smaller than the input.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int isprime(int n) {
  int i,squareroot;
  if (n>10) {
    squareroot = (int) sqrt(n);
    for (i=3; i<=squareroot; i=i+2)
      if ((n%i)==0)
        return 0;
    return 1;
  }
  else
    return 0;
}

int main(int argc, char *argv[])
{
  int pc,       /* prime counter */
      foundone; /* most recent prime found */
  long long int n, limit;

  sscanf(argv[1],"%llu",&limit);
  printf("Starting. Numbers to be scanned= %lld\n",limit);

  pc=4;     /* Assume (2,3,5,7) are counted here */

  for (n=11; n<=limit; n=n+2) {
    if (isprime(n)) {
      pc++;
      foundone = n;
    }
```

```
  }
  printf("Done. Largest prime is %d Total primes %d\n",foundone,
     pc);

  return 0;
}
```

2 Statement of Problem 2

In this problem, you need to use MPI to parallelize the following serial program (`http://www.cs.nctu.edu.tw/~ypyou/courses/PP-f17/assignments/HW3/integrate.c`), which integrates function sin(X) over the range from 0 to π using `N` intervals, where `N` is an argument of the program.

```
#include <stdio.h>
#include <math.h>

#define PI 3.1415926535

int main(int argc, char **argv)
{
  long long i, num_intervals;
  double rect_width, area, sum, x_middle;

  sscanf(argv[1],"%llu",&num_intervals);

  rect_width = PI / num_intervals;

  sum = 0;
  for(i = 1; i < num_intervals + 1; i++) {

    /* find the middle of the interval on the X-axis. */

    x_middle = (i - 0.5) * rect_width;
    area = sin(x_middle) * rect_width;
    sum = sum + area;
  }

  printf("The total area is: %f\n", (float)sum);

  return 0;
}
```

3 Requirement

- Your submitted solution contains two source files: `prime.c` (or `prime.cpp`) for problem 1 and `integrate.c` (or `integrate.cpp`) for problem 2.
- Your programs take one command-line argument.

4 Developing and Execution Environment of MPI Programs

4.1 Using the NCTU CS virtual cluster

4.1.1 Login information

IP of the master workstation: pp1.cs.nctu.edu.tw
User name & password: The same account and password as NCTU CSCC account(`https://www.cs.nctu.edu.tw/cscc/account/`)
For more information, please refer to NCTU CS Computer Center (`https://www.cs.nctu.edu.tw/cchonor/`)

4.1.2 You can use SSH to log in to other slave machines

Hostname	IP
host	pp1.cs.nctu.edu.tw
slave1	pp2.cs.nctu.edu.tw
slave2	pp3.cs.nctu.edu.tw
slave3	pp4.cs.nctu.edu.tw

Try the command below to see whether the command exist or the PATH variable is properly set.

```
$mpiexec --version
or
$echo "$PATH"
```

4.2 Executing jobs on slaves without entering a password

To make `mpiexec` work properly, you need to be able to execute jobs on remote nodes without typing a password. You will need to generate a ssh key by yourself.

You can also google "ssh passphrase" for detail.

```
//Please create .ssh directory if not exists
user@host:~$ mkdir -p ~/.ssh
user@host:~$ ssh-keygen -t rsa
//Then you will be prompt to enter a passphrase,you can leave it
   empty.(Empty is more conveniet)
user@host:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

```
//All node share with same directory, so copying to host will
    also work on slaves
```

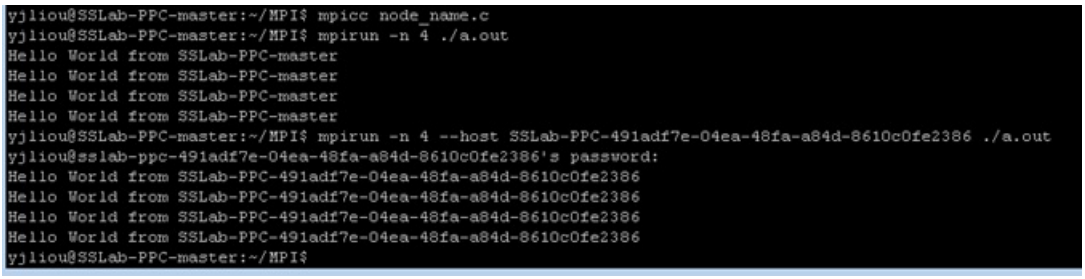
Next, loginning to all slaves at least one time to finsih configuration.

If you were on host, you can try the following command for loginning without password to see whether the configuration works.

```
$ssh slave1
or
$ssh slave2
or
$ssh slave3
```

4.2.1 Writing your program and running

You have to copy your executable to each of the slave machines, and the file location should be the same as the location on your master machine.(Use the hostname defined above.) Note: The NCTU CSCC servers have a NFS and NIS system, so you don't need to copy file by yourself. You will see the same filesystem on the four servers.



4.3 Compiler and mpiexec flag for the homework

You may need to use `mpicc` with the `-lm` flag to link properly. Here is an example for how to compile and run an MPI program (taking "integrate" as an example):

```
$mpicc -o integrate integrate.c -lm
$mpiexec -n 4 -host host,slave1,slave2,slave3 ./integrate
   123456789
```

5 Submission

Be sure to upload your zipped source codes, which includes no folder, to e-Campus system by the due date and name your file as "HW3_xxxxxxx.zip", where xxxxxxx is your student ID.

Due Date: 23:59, December 1, Friday, 2017

6 References

- https://computing.llnl.gov/tutorials/mpi/