Parallel Programming
Fall 2017/2018
programming assignment 2
Due by 1/10/2017
Adnan Salman

Objective: The goal of this assignment is to get you start programming distributed memory systems using MPI.

In this programming assignment, you will consider a simple algorithms to find the prime numbers between 1 up to some integer N (N is given as input). Simply, by checking all divisors

```
int isprime ( n )
    for j=2 to sqrt(n)
        if ( n % j == 0)
            return false // not prime
    return true  // prime
```

Q1: Why the algorithms checks only up to sqrt(n) not up to N?

Since all prime numbers greater than 2 are odd numbers, we can ignore all even numbers

```
if (n > 2 )
    print  2 is a prime number
for  i = 3 to n step by 2
    if ( isprime (i) )
        print  i is a prime number
```

Q2: Write a C++ serial program primenum.cpp that takes an integer N as argument and prints all primes number up to and includes N. Sample usage of your program should be

```
$ ./primenum 11
  2  3  5  7  11
$./primenum 12
  2  3  5  7  11
```

Use unsigned long int, so your program can handle large numbers. Also, remember how to use argc and argv parameters of the main function to get the command line arguments

Now, you need to parallelize the computation of the prime numbers, i.e, the code

```
for  i = 3 to n step by 2
    if ( isprime (i) )
        print  i is a prime number
```

using MPI. You can do that by assigning the first block of the loop to process 0, the second block to process 1, and so on. For example, if N = 32, and you run the program with 3 processes, then the assignment looks like

```
process 0:   3, 5, 7, 9, 11
process 1:   13, 15, 17, 19, 21
process 2:   23, 25, 27, 29, 31
```

Q3: Write an MPI program mpiprime.cpp that display the prime numbers between 1 and N using the above assignment of work to processes. It is ok that the prime numbers be printed not in order. We can deal with printing them in order in another assignment. Example, usage

```
$ mpirun -np 3 ./mpiprime 32
process 0:   3, 5, 7, 11    ( time = seconds )
process 1:   13, 17, 19     ( time = seconds )
process 2:   23, 29, 31     ( time = seconds )
```

Q4: In the above assignment, it is easy to see that there is a load imbalance, process 0 gets the smallest numbers while process p-1 gets the largest numbers. We can improve our solution by using cyclic assignment instead. In this assignment, processes get roughly same amount of small numbers and large number. Using cyclic assignment in the above example, the numbers should be distributed as

```
process 0:    3,  9,  15, 21, 27
process 1:    5 ,  11, 17 ,23, 29
process 2:    7,   13,  19 ,25,  31
```

Now, rewrite the mpiprime program using cyclic distribution call it mpiruncyclic.cpp. Did you get better performance.

Q4: Run the program using different number of process up to number of cores in your machine. Discuss the speed up over the serial code of the program.

How to time a block of your program?

Use gettimeofday function if you are running serial program without MPI

```
#include <sys/time.h>
double GetWallTime() {
    struct timeval tp;
   int rtn = gettimeofday(&tp, NULL);
   return ((double) tp.tv_sec + (1.e-6)*tp.tv_usec);
}

int main()
   double start = GetWallTime();
```

```
    // Code to be timed /
    double finish = GetWallTime();
    cout  << "Elapsed time = " << finish – start << endl;
}
```

Also, you can MPI_Wtime() function from MPI for MPI application

What to turn in?

You should turn in by 20/9/2015, using dropbox the following files

1) The serial program prime.cpp
2) The mpiprime.cpp
3) The mpiprimecyclic.cpp
4) A write up the include the following
    1) Description of what you did
    2) Answer to Q1
    3) pseudocode for the mpiprime.cpp code, the pseudocode should just explain how did you make the assignments of the iterations to process
    4) pseudocode for the mpiprimecyclic.cpp code, the pseudocode should just explain how did you make the cyclic assignments of the iterations to process
    5) Discuss the results you obtained.

Work in team?

It is strongly recommended to discuss the assignment with your classmates. However, any code copying will be considered cheating.