Assignment 3

Q1. DAXPY Loop

D stands for Double precision, A is a scalar value, X and Y are one-dimensional vectors of size 2^16 each, P stands for Plus. The operation to be completed in one iteration is X[i] = a*X[i] + Y[i]. Implement an MPI program to complete the DAXPY operation. Measure the speedup of the MPI implementation compared to a uniprocessor implementation. The double MPI Wtime() function is the equivalent of the double omp get wtime().

Q2. Calculation of π - MPI Bcast and MPI Reduce

This is the first example where many processs will cooperate to calculate a computational value. The task of this program will be arrive at an approximate value of π. The serial version of the code follows.

```
static long num_steps = 100000;

double step;

void main ()

{

int i;

double x, pi, sum = 0.0; step = 1.0/(double)num_steps; for (i=0; i<num_steps; i++) { x = (i+0.5)*step;

sum = sum + 4.0/(1.0+x*x);

}

pi = step * sum;

}
```

Your task is to create a parallel MPI version of the π program. For this version, use the MPI Bcast and MPI Reduce functions.

• Broadcast the total steps value (num steps) to all the processs (process 0 could broadcast). The syntax and usage of MPI Bcast follow:

– MPI Bcast(void *message, int count, MPI Datatype datatype, int root, MPI Comm comm);:

Broadcast a message from the process with rank “root” to all other processes of the group. – Example: MPI Bcast(&n, 1, MPI INT, 0, MPI COMM

WORLD);. process 0 broadcasts n, of type MPI INT, to all the processs MPI COMM WORLD.

- Each process calculates its partial π value.
- Use MPI Reduce to reduce the partial π values generated by each process. MPI Reduce is executed by every process.

Q3. Write a program to find all positive primes up to some maximum value, using MPI_Recv to receive requests for integers to test. The master will loop from 2 to the maximum value on

1.issue MPI_Recv and wait for a message from any slave (MPI_ANY_SOURCE), 2.if the message is zero, the process is just starting, if the message is negative, it is a non-prime, if the message is positive, it is a prime.

3.use MPI_Send to send a number to test.

and each slave will send a request for a number to the master, receive an integer to test, test it, and return that integer if it is prime, but its negative value if it is not prime.