ITCS 4/5145 Parallel Computing
Test 1 5:00 pm - 6:15 pm, Wednesday February 17, 2016

Name: .....

Answer questions in space provided below questions. Use additional paper if necessary but make sure your name is on additional sheets. *Clearly show how you obtained your answers. (No points for simply writing a numeric answer without showing how you got the answer, even if correct.)* This is a closed book test. Do not refer to any materials except those supplied for the test. Supplied: *"Summary of OpenMP 3.0 C/C++ Syntax." and "Basic MPI routines."*

Total /40
7 pages

Qu. 1 Answer each of the following briefly:

(a) According to Amdahl's law, what the maximum speed-up of a parallel computation given that 90% of the computation can be divided into parallel parts? What assumption are you making? 2

(b) How can one make five threads do exactly the same code sequence using OpenMP?

2

(c) In the following OpenMP code sequence, which variable or variables must be declared as private variables (in a private clause):

```
int x, tid, a[100];
#pragma omp parallel
{
    tid = omp_get_thread_num();
    n = omp_get_num_threads();
    a[tid] = 10*n;
}
```

2

(d) What does the nowait clause do in the Open sections directive? 2

(e) What is the value of sum after the following OpenMP code sequence, i.e. what number does the printf statement print out? 2

```
int i, sum;
omp_set_num_threads(2);
sum = 0;

#pragma omp parallel for reduction(+:sum)
for (i = 0; i < 5; i++ ) {
    sum++;
}
printf("Sum = %d\n", sum);
```

(f) If x is a shared variable initialized to zero and three concurrent threads execute the statement x = x + 1; what are the possible value of x afterwards? 2

(g) Identify all the dependencies in the following sequence:

1. **a = b + c;**
2. **y = a;**
3. **a = 3;**
4. **x = y + z;**

Clearly show how you got your answer. The statements are numbered so that you can refer to them. 2

(h) In the MPI statement:

MPI_Send(&x,1,MPI_INT,1,msgtag, MPI_COMM_WORLD);

what can be inferred about how x has been declared. 2

(i) In MPI, how does the programmer specify how many processes the program will use? 2

(j) When does the MPI routine MPI_Send() return? 2

(k) What is a Jacobi iteration? 2

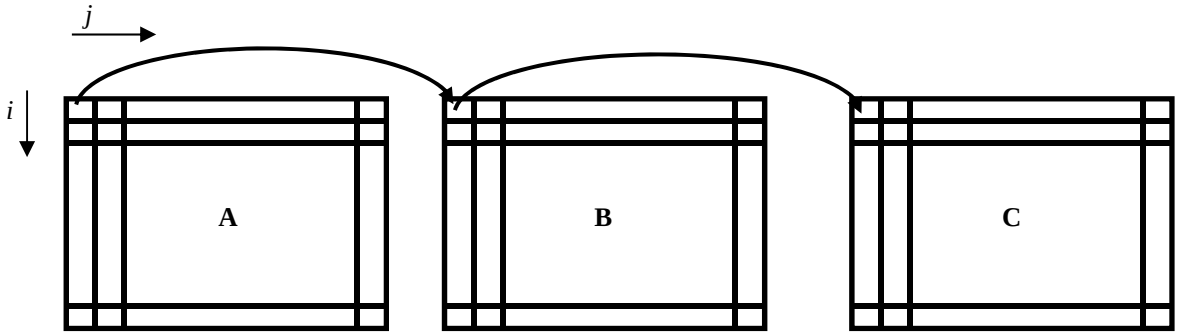
(l) What does the -l option specify when used with the Linux C compiler (gcc or cc)? Give an example. (The -l option was used in Assignment 2, but the question is asking what -l specifies in general.) 2

The following include statements can be assumed:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <omp.h>
#include "mpi.h"
```

in Qu 2 a, b and c. However any define statements, variables, and arrays that you use must be declared.

Qu. 2 (a) Write a sequential C program to perform matrix addition adding two matrices **A[N][N]** and **B[N][N]** to produce a matrix **C[N][N]**. The arrays hold doubles. **N** is a constant defined with a define statement and set to 256. In matrix addition, the corresponding elements of each matrix are added together to form elements of result matrix, as shown below:



i.e. given elements of **A** as $a_{i,j}$ and elements of **B** as $b_{i,j}$, each element of **C** computed as $c_{i,j} = a_{i,j} + b_{i,j}$. You can assume that the arrays are initialized with values but show where that would be in the program with comments.

4

(b) Modify the program in 2(a) become an OpenMP program performing matrix addition using ***T*** threads where ***T*** is a constant defined with a define statement and set to 16.

4

(c) Modify the program in 2(a) to become an MPI program performing matrix addition using ***P*** processes where ***P*** is determined when the code is executed. Partition the matrices into **blksz** rows, where **blksz** = ***N***/***P*** (Clue: remember how matrix multiplication was done.) It can be assumed that ***N***/***P*** is an integer.

8