ITCS 4145/5145 Parallel Programming
Final exam
Tuesday December 11th, 2012, 11:00 am - 1:30pm

Name: .....

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.*"

Answer questions in space provided below questions. Use additional paper if necessary but make sure your name is on every sheet.

Total /60

Do not refer to any materials for this part

Qu. 1 Answer each of the following briefly:

(a) According to Amdahl's law, what is the maximum speed-up of a parallel computation given that 80% of the computation can be executed in parallel? Clearly explain. No points for simply putting down a numerical answer with an explanation. 2

(b) Which computer system was used at UNC-Charlotte for OpenMP programming assignments? Why? 2

(c) In the Seeds framework, the programmer uses the DataMap class, which extends the Java Hashmap class. How are data items identified within objects of this class? 2

(d) Which IDE (Integrated Development Environment) was used in Assignment 1? 2

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

(f) What does the MPI routine MPI_Gather() do? 2

(g) How does one specify that a section of code should be run in parallel on available processors using the Paraguin compiler? Give a specific example. 2

(h) Write for Paraguin pragma to execute blocks of iterations of the following for loop in parallel: 2

```
for (i = 0; i< n; i++)
    …
```

(i) Write the Paraguin pragma to get the input data array float a[n] to all the processors. 2

(j) In the instruction to generate graphical output for Assignment 4, it is suggested that you will need a sleep() statement in the code. Why? What does this statement do? 2

(k) If we have an algorithm whose sequential complexity is O(*N*) and we parallelize it with *P* processors, what is the maximum speedup we would expect (write this as a formula)? Is it possible to achieve greater speedup and, if so, under what circumstance might this happen? 2

(l) What arithmetic property is required for an operation applied as a reduction? 2

(m) Under what circumstances would the message tag be imperative in MPI? 2

(n) Describe how sorting could be done using a pipeline pattern 2

(o) If two threads execute the instruction x++ where x is a shared variable initialized to 0, what are the possible values that x could have after the execution of the threads? Clearly explain your answer. 2

(p) How many threads could be used for the computation below, each thread executing one or more of the instructions: 2

```
x++ ;
a = x + 2;
b = a + 3;
c++;
```

without changing the code. Explain clearly your answer.

Challenge: How could the code sequence be changed to expose more parallelism but still achieve the same final result?

2

(q) If a routine is *sequentially consistent* when the same routine is executed on separate threads at the same time, why can we say the routine is *thread-safe*? 2

(r) Many sorting algorithms use a compare an exchange operation. Suppose the compare and exchange operation operates on two groups of sorted numbers rather than individual numbers. Describe how the compare and exchange is achieved. (Just one way is needed) 2

(s) Which pattern is used in GPU programming with CUDA? 2

Qu. 2 Write a *complete* OpenMP program that determines all the primes from 2 to N^2 using Sieve of Eratosthenes. Share the work across N OpenMP threads.

Provide very clear explanation of how the program works, and comments in your code. If I do not understand the code, I will assume it is incorrect.

10

Qu 3 Write a *complete* CUDA program to add two *N* x *N* matrices, **A** and **B**, where *N* is a constant. Use #define statement for *N*. Use one thread for adding each pair of elements within square 2-dimensional blocks and a square 2-dimensional grid structure. The number of threads in a block and number of blocks in the grid (in each dimension) are to be given as #define constants, *T* and *B* respectively. The code is to work if *T* x *B* is greater than *N*. Initialize each matrix with numbers (any numbers) and print out the final results.

Some CUDA predefined variables:

The thread ID within a block in the x direction is given by threadIdx.x
The block ID in the x direction is given by blockIdx.x
The number of bocks in the x direction is given by blockDim.x

Provide very clear explanation of how the program works, and comments in your code. If I do not understand the code, I will assume it is incorrect.

10