CSCI-UA.0480-003

Parallel Computing

Final Exam

Spring 2014 - May 15th (90 minutes)

NAME: **ID:**

- This exam contains 8 questions with a total of 20 points in **four pages.**
- The exam is open book/notes.
- If you have to make assumptions to continue solving a problem, state your assumptions clearly.
- You answer on the question sheet. You can use extra white papers if you want.

1. [2 points] If three threads, in OpenMP, 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 (assume no synchronization or precautions were taken)? Clearly explain how each value(s) results.

2. [2 points] When is loop-unrolling **not** beneficial? Assume we are talking about CUDA.

3. [1 point] Suppose we have a system with 1 CPU and 4 GPUs. The CPU can reach a performance of 0.5 GFLOPS, while each GPU can reach 1GFLOPS. If you have an application which is 50% parallelizable (i.e. 50% of the application is assigned to CPU and the other 50% to the GPUs), what is the average peak performance?

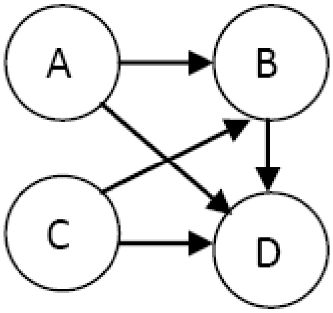
4. [4 points] Parallelize the following code using openMP pragmas. Be sure to explicitly specify the "schedule" options that should be used, even if you want to use the default options. You can assume that the variable P represents the number of processors to be used. Also assume that N is large (in the tens of thousands or more).

```
C[0] = 1;
for (i=1;i<N;i++){
        C[i] = C[i-1];
        for (j=0;j<N;j++){
                C[i] *= A[i,j] + B[i,j];
        }
}
```

5. [4 points] In bulleted list state the source(s) performance loss that we may face in: MPI, OpenMP, and CUDA (a bulleted list for each).

6. [3 points] Explain whether we can have race condition in MPI, OpenMP, and CUDA, and justify in each case.

7. [2 points]Assume that all instructions of an application can be partitioned into 4 groups (A, B, C, D), with the following dependency. Each group contains 25% of the instructions. How to schedule them on your system (CPU + 4 GPU coprocessors) to achieve the best possible performance? Assume an instruction group can be assigned to either a CPU or GPU (Hint: Pay attention to communication cost!)



8. [2 points] Beside overlapping data-transfer and computation, state two other scenarios where streams are useful in CUDA.