CSCI-UA.0480-003

Parallel Computing

Midterm Exam

Spring 2016 (60 minutes)

NAME: **Net ID:**

- This exam contains **7 questions** with a total of 20 points in **4 pages.**
- You are allowed only 1 paper (2 pages) of notes.
- **If you have to make assumptions to continue solving a problem, state your assumptions clearly.**

1.[1 pt] Describe a scenario, no need to write code, where having two MPI processes can still yield better performance than sequential program on a single-core processor.

2.[1 pt] What is the main reason for moving from single core to multicore processors?

3. [2 pts] Suppose we have two MPI processes. Describe a scenario, no need to write code, where they would execute slower on a system with two cores than on a system with one core.

4.[2 pts] How could the following code sequence be changed to expose more parallelism but still achieve the same final result (i.e. at the end: x, a, b, and c have the same value as the sequential code)? In your solution, show the largest number of parallel tasks and what each task will do. Disregard any overheads, just show the parallelism. The optimal solution must not contain extra work than the original sequential code.

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

5. [4 pts] In MPI, explain two ways where a process can send data to a *subset* of the processes in MPI_COMM_WORLD.

-

-

6. [6 pts] The following code snippet has one or more bug(s). Show and correct this (or those) bugs. No need to rewrite the code. Just state the bug and how you will fix it.

```
double a[100], b[100];

MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &myrank);

if( myrank ==0){
        MPI_Recv( b, 1000, MPI_INT, 1, 19, MPI_COMM_WORLD, &status );
        MPI_Send( a, 1000, MPI_INT, 1, 17, MPI_COMM_WORLD );
        }
else if( myrank ==1){
        MPI_Recv( b, 1000, MPI_INT, 0, 17, MPI_COMM_WORLD, &status );
        MPI_Send( a, 1000, MPI_INT, 0, 19, MPI_COMM_WORLD );
        }
MPI_Finalize();
```

7. In class we have seen that two overheads are load imbalance and synchronization. In the following two questions, disregard any overheads except synchronizations and load imbalance.

(a)[2 pts] If we, magically, eliminate load imbalance from our code, do synchronization points (i.e. barriers) still affect performance? Justify.

(b)[2 pts] Since we cannot guarantee that we eliminate load imbalance, we decide to eliminate the barriers from our code. Does load imbalance still affect performance? Explain in 1-2 sentences only.