CSC 478 Winter 2010 final exam. Due date: Friday April 23, by 1:00 p.m.

This exam is worth 20% of your course grade, graded on a 100-point scale (all questions are weighted equally). The exam should be submitted in a document through the file upload or emailed. Typeset (Word, PDF) is preferred, but (scanned) hand-written is also acceptable as long as it is legible.

1. In transactional memory systems, a thread optimistically assumes that it makes no references to shared data that conflict with other accesses by other threads. The transaction either commits successfully if there was no access violation detected, or the transaction rolls back if there was. Identify the sources of performance loss in a transactional memory system, classifying each as overhead, contention, or idle time.

2. Consider the arithmetic expression $a + b/c - d + e((f + g/h)/x + y)$. Using associativity, commutativity, and distributivity, transform it so that it can be evaluated in minimum time. Show the resulting expression tree and find the minimum number of processors needed for the minimum time evaluation.

3. Explain superlinear speedup: what it is, including the various possible causes/sources. (Cite references where appropriate.)

4. Consider the problem of adding n numbers by using n processing elements. Initially, each PE is assigned one of the numbers to be added and, at the end of the computation, one of the PEs stores the sum of all the numbers. Describe an algorithm for doing this in parallel.

5. Suppose that in a certain computation, 80% is fully vectorizable. Of the remaining sequential part, up to 60% of it may also be fully parallelized for an MIMD machine. What are the speedups over sequential execution for a 10 PE SIMD machine and a 10 processor MIMD machine for this computation?

6. Consider the following two sequential code segments:

   ```
   sum = a + 1;
   first = sum * scale1;
   sum = b + 1;
   second = sum * scale2;
   ```

   ```
   first = (a + 1) * scale1;
   second = (b + 1) * scale2;
   ```

   If there are any, identify what type of dependencies exist in the code.

7. The following parallel MIMD code computes the histogram of a data array. x0 and x1 are the minimum and maximum values, respectively, in the data array x[1:N], and the histogram vector is h[0:M-1]. The function floor truncates a floating-point number:

```
  shared scale, M, x1, x0, x[N], h[M];
  private i, j;
  scale = M/(x1 - x0);
  for i = 1 to N-1 fork DOELEMENT;
  i = n;

DOELEMENT:
  j = floor(scale*(x[i] - x0));
  critical
    h[j] = h[j] + 1;
  end critical
  join(N);
```

Convert this to a program that uses a a defined procedure called with **call** and **create** instead of fork/join.

8. Three processors in a distributed memory multiprocessor communicate using send and receive, running according to the sketch below.

   (a) Draw a diagram of the precedence relation on local activities if send is nonblocking and receive is blocking.

   (b) Given part (a), identify the longest chain of activities that must be executed in sequence.

time

Process P1	**Process P2**	**Process P3**
A	E	G
send(P2)	receive(P3)	receive(P1)
B	E	H
send(P3)	receive(P1)	send(P2)
C	F	I

9. Explain the differences among semaphores, low-level locks, critical/end critical, and the produce/consume constructs.