EECS 739: Exam # 1

Thursday, March 12, 2015

Print Name and Signature________________________________

The rules for this exam are as follows:

- **Write your name on the front page of the exam booklet. Initial each of the remaining pages in the upper-right hand corner. Sign the front of the exam booklet. Your exam will not be graded if you have not signed the front page of the booklet.**
- The exam has 4 questions and 1 extra credit question. The exam is 7 pages long (including this page). Be sure you have all of the pages before beginning the exam.
- This exam will last for 75 minutes.
- Show **ALL** work for partial/full credit. This includes any definitions, mathematics, figures, etc.
- The exam is closed book and closed notes.
- You are allowed to use a calculator on the exam provided you use it to perform arithmetic computations only and show all of your work on the exam.
- No collaboration of any kind is allowed on the exam.

1. _______ (20 points)	4. _______ (20 points)
2. _______ (20 points)	EC. ______ (10 points)
3. _______ (20 points)	T. _______ (80 points)

1. (20 points; 2 points each) If possible, give an example of each of the following. If not possible, write "NOT POSSIBLE". **Do NOT include any additional justification.**

   (a) A parallel programming language that is used with distributed memory machines.

   (b) A task-dependency graph for a parallel code that results in guaranteed deadlocks.

   (c) A situation in which superlinear speedup can occur.

   (d) Three categories of serial linear solvers we covered in class.

   (e) A linear system of the form with $Ax = b$, where A is 3×3, x and b are column vectors of length 3, and the system causes instabilities during the numerical solution procedure.

   (f) One criterion used to evaluate the cost and performance of static network topologies.

(g) Two aspects of the cluster and the associated network for which you wrote software to measure them.

(h) An example of a problem in which exploratory decomposition is used.

(i) A linear solver we studied in class which has a complexity greater than that of Gaussian Elimination.

(j) An MPI command which is used in the core of a parallel backward substitution algorithm assuming the n rows of A (and corresponding entries in b) are owned by n different processors.

2. (20 points; 4 points each) Consider a processor operating at 2 GHz connected to a DRAM with a latency of 50 ns. Assume the processor has two multiply-add units and is capable of executing four instructions in each cycle. Suppose there is also a cache of size 32 KB with a latency of 2 ns. Use this architecture to multiply two matrices A and B of dimensions 32×32. Note that A and B, as well as $C = A * B$ fit in the cache. Assume an ideal cache placement strategy in which none of the data items are overwritten by others.

   (a) How many words need to be fetched into the cache, i.e., in order to fetch A and B into the cache? How long does this take?

   (b) How many operations are performed in order to multiply the two matrices? How long does this computation take?

   (c) What is the total time (i.e., for fetching and performing computation)?

   (d) What is the peak computation rate?

   (e) List three different approaches to hiding memory latency. **Note: This last subquestion is a general one.**

3. (20 points) Consider solving the linear system $Ax = b$, where

$$A = \begin{pmatrix} \tilde{A}_1 & & & & & \\ & \tilde{A}_2 & & & & \\ & & \tilde{A}_3 & & & \\ & & & \tilde{A}_4 & & \\ & & & & \ddots & \\ & & & & & \tilde{A}_m \end{pmatrix} \tilde{x} = \tilde{b},$$

where $\tilde{x} = Mx, \tilde{b} = Mb$. In addition, $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_m$ are blocks of size $\frac{n}{m} \times \frac{n}{m}$ and $M = n \times n$. The remainder of A is composed of zeros.

(a) (15 points) Write a pseudocode for a parallel algorithm which solves the above linear system. **Be sure to specify which MPI functions you would use for each step. You do NOT need to write actual MPI code.**

(b) (5 points) Specify a real-world application for which linear systems of the above form arise. **(Hint: $M = F_b{}^*$ which was an inverse discrete Fourier transform in lecture. Here we use generic matrices M.)**

4. (20 points; 4 points each) Consider an algorithm with problem size, $W = n^2$, serial run time, $T_s = n^2 t_c$, and parallel run time, $T_p = t_c(n^2/p) + t_s \log p + t_w n$, where t_c is the amount of time taken to perform a unit of computation, t_s is the start-up time of the communication network, and t_w is the per-word transfer time.

   (a) Determine an expression for the speedup, S.

   (b) Determine an expression for the efficiency, E.

   (c) Determine an expression for the total overhead, T_0.

   (d) Determine the isoefficiency function for this algorithm.

   (e) Explain how to interpret the isoeffiency result.

OPTIONAL: Extra-Credit Question

1. (10 points) Recall that the Cholesky factorization algorithm is used to factor symmetric positive definite matrices A into $A = LL^T$, where L is lower triangular.

   A generalization of the Cholesky factorization is the LDL^T factorization. Here, A is factored as follows: $A = LDL^T$, where L is unit lower triangular, and D is diagonal.

   (a) (6 points) To determine how the LDL^T factorization algorithm is a generalization of the Cholesky factorization algorithm, consider computing both the Cholesky and LDL^T factorizations of the following matrix:

   $$A = \begin{pmatrix} -4 & 0 & -8 \\ 0 & -2 & -12 \\ -8 & -12 & -85 \end{pmatrix}.$$

   (b) (2 points) Explain the generalization.

   (c) (2 points) Besides being a generalization of the Cholesky factorization algorithm, there is a distinct advantage of the LDL^T factorization algorithm. What advantage did you observe when computing the factorizations of the above matrix?