Fall 2011

High-Performance Computing

1. **Work, depth (span), and parallel prefix.** Give an efficient parallel algorithm for computing the first n Fibonacci numbers, whose sequential pseudocode is as follows.

   1: $F_0 \leftarrow 0$
   2: $F_1 \leftarrow 1$
   3: **for** $i = 1$ to n **do**
   4: $\quad F_{i+1} \leftarrow F_i + F_{i-1}$
   5: **end for**

   Analyze the work $W(n)$ and depth (span) $D(n)$ of your algorithm. Then, use this analysis to estimate the running time of your algorithm on p processors.

2. **Algorithms for manycore processors.** Consider a manycore processor whose cores are laid out in a 2D mesh of size $\sqrt{p} \times \sqrt{p}$. Further suppose that each core has a private local-store of size Z words. Give an efficient parallel algorithm for performing matrix multiply on $n \times n$ matrices, where $n^2 \gg p \cdot Z$. Analyze the communication costs of your algorithm, making the following assumptions:

   (a) The time to transfer a block of m words from main memory to the processor is $\alpha_{\text{mem}} + m/\beta_{\text{mem}}$. Assume that this time assumes the time to distribute the m words among all of the p local-stores.

   (b) The time to transfer a block of m words from one core to any of its nearest-neighbors in the mesh is $\alpha_{\text{net}} + m/\beta_{\text{net}}$.

3. **Loop transformations.**

   (a) (30%) Consider the following loop.

   1: {Let A and B be two arrays, each of size m floating-point elements}
   2: **for** $i = 1$ to n **do**
   3: $\quad A[i] \leftarrow B[k] + A[i] + 10^6$
   4: **end for**

   Suppose a compiler transforms this loop into the following:

   1: $t \leftarrow B[k] + 10^6$
   2: **for** $i = n$ to 1 by steps of -1 **do**

3: $A[i] \leftarrow A[i] + t$
4: **end for**

Explain why this transformed code may produce different results from the original. (Hint: There are multiple reasons; for full credit, identify at least two.)

(b) (70%) Consider the following sequential loop nest.

1: **for** $i = 1$ to n **do**
2: $Z[i] \leftarrow Z[i]/W[i]$
3: **for** $j = i$ to n **do**
4: $X[i,j] \leftarrow X[i,j] * Y[i,j]$
5: $Z[j] \leftarrow Z[j] + X[i,j]$
6: **end for**
7: **end for**

Rewrite this loop nest to maximize the opportunities for vectorization. Describe (at a high-level) the assumptions and the sequence of analysis and transformation steps that a compiler would need to carry out to arrive at your loop nest.

4. **Shared memory abstractions on a distributed memory platform.** You are designing a runtime library that will support the implementation of a shared memory programming model "reasonably" efficiently on a distributed memory platform. In about one to two pages, please answer the following questions.

(a) (30%) What are the most important characteristics of a shared memory programming model relevant to your task? Try to come up with at least three such characteristics.

(b) (30%) What are the major characteristics of a distributed memory platform relevant to your task? Try to come up with at least three such characteristics.

(c) (40%) Explain the challenges that your answers to (b) pose for the characteristics you identified in (a).

Fall 2011

Numerical Methods

1. Let A denote a nonsingular matrix with n rows and columns. Let E denote a perturbation matrix such that $A + E$ is singular. For any induced matrix norm, show that

$$\kappa(A) \geq \frac{\|A\|}{\|E\|}$$

where $\kappa(A)$ is the condition number of A. This result is consistent with the fact that matrices that are close to being singular ($\|E\|$ is small) have large condition numbers.

2. Let $A(y)$ denote a symmetric, positive definite matrix function of dimension n and let y denote an n-by-1 vector function of time, t. Let $f(t)$ denote a given function. Consider the ordinary differential equation

$$A(y)\frac{dy}{dt} = f(t)$$

with the initial condition $y(0) = y_0$.

   (a) What can be said about the exact solution, $y(t)$?

   (b) Write the first step of the explicit Euler method for solving this equation, with step size h.

   (c) What is the maximum stable step size? Express your answer in terms of A.

   (d) Write the first step of the implicit Euler method.

   (e) What are the nonlinear equations to be solved in the implicit Euler method, and what are your recommendations for solving these nonlinear equations?

3. In this question, you are asked to write an efficient algorithm for computing the cube root $y^{1/3}$ of any given positive number y. Your algorithm must be designed for a base 2 floating point system, where any

floating point number can be represented as $y = a \times 2^e$, where a is a normalized fraction ($0.5 \leq a < 1$) and e is an integer exponent. For efficiency, you may store some useful constants ahead of computation time, e.g., $2^{1/3}$, $2/3$, and $a/3$, assuming these are useful.

(a) Show how $y^{1/3}$ can be obtained once $a^{1/3}$ has been calculated for the corresponding fraction, in as little as five additional flops.

(b) Derive the corresponding Newton iteration. What is the flop count per iteration?

(c) How would you choose an initial approximation? Roughly how many iterations are needed if the machine rounding unit is 2^{-52}?

4. Consider the two-point boundary value problem

$$\frac{d^2u}{dx^2} + u = -x, \quad 0 \leq x \leq 1$$

with boundary conditions $u(0) = 0$ and $u(1) = 0$. Consider solutions of the form

$$v(x) = x(1-x)(a_1 + a_2 x)$$

where a_1 and a_2 are parameters to be determined.

(a) What is a general advantage of the Finite Element Method over the Finite Difference Method for solving two-point boundary value problems (or partial differential equations in general)?

(b) Use the method of weighted residuals to solve the boundary value problem. Use the weight functions $w_1(x) = x(1-x)$ and $w_2(x) = x^2(1-x)$.

(c) Use collocation to solve the boundary value problem. Use the two collocation points $t_1 = 0.5$ and $t_2 = 0.75$. Do not use the boundary conditions when constructing equations to solve. Why are the boundary conditions not necessary in your solution method?

Fall 2011

Data Analysis

1. Let $\phi(y;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$ denote the density of a random variable y with a Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$. Suppose that we have three related random variables, X, Y and Z,

   - Random variable X has a Gaussian distribution $\mathcal{N}(0,\sigma^2)$;
   - Given random variable $X = x$, random variable Y has a Gaussian distribution $\mathcal{N}(x,\sigma^2)$;
   - Given random variable $Y = y$, random variable Z is a mixture of two Gaussians with density

   $$p(z|Y=y) = (1-\alpha)\phi(z;0,\sigma^2) + \alpha\phi(z;y,\sigma^2). \tag{1}$$

   - Conditioned on random variable Y, random variables X and Z and independent

   Given n i.i.d. sample $z^1,\ldots,z^n$ from the mixture density (1), answer the following questions

   (a) If $n = 2$, derive the posterior distribution of X conditioned on $(z^1,\ldots,z^n)$ exact upto a scalar difference.

   (b) If $n = 10$ (or in general when n is large), what is the computational problem associated with computing the poterior distribution of X?

   (c) Propose approximation algorithms to deal with the computational problem when n is large.

2. Let $X_1,\ldots,X_n$ be n determinations of a physical constant θ. Consider the model,

   $$X_i = \theta + e_i, \quad i = 1,\ldots,n$$

   and assume

   $$e_i = \alpha e_{i-1} + \beta e_{i-2} + \epsilon_i, \quad i = 1,\ldots,n, \quad e_0 = 0, e_{-1} = 0$$

   with ϵ_i's iid standard normal, and α and β are known constant. What is the maximum likelihood estimate of θ? Carefully justify each step of your derivation/calculation.

3. Consider two machine learning models for 2-class classification. The first is a support vector machine with Gaussian kernel. The second is kernel discriminant analysis (a Bayes classifier with a kernel density estimator for each class), where the bandwidth may vary for each dimension, and possibly also for each data point. Which is the more expressive, or powerful model? Compare and discuss the pros and cons of each.

4. (a) Write down the Bayes classifier $f : X \to Y$ (the classifier that minimizes the expected loss $E(L(Y,f(X)))$) for binary classification $Y \in \{-1,+1\}$ with non 0-1 loss (a is the loss for falsely predicting negative and b is the loss for falsely predicting positive). Simplify the classification rule as much as you can.

(b) If $P(X|Y = y)$ is a multivariate Gaussian and assuming the 0/1 loss, write the Bayes classifier as $f(X) = \text{sign}(h(X))$ and simplify h as much as possible. What is the geometric shape of the decision boundary?

(c) Repeat (b) when the two Gaussians have identical covariance matrices. What is the geometric shape of the decision boundary?

(d) Repeat (b) when the two Gaussians have covariance matrix that equals the identity matrix. Describe the geometric shape of the decision boundary as much as possible.

Fall 2011

CSE Algorithms

1. In computational biology, DNA can be represented as a sequence of characters drawn from an alphabet of four letters, A, C, T, and G, representing the four nucleotides. Given two sequences S_1 and S_2 of n and m characters, respectively, describe what is meant by a local alignment. Given a similarity score of +2, a mismatch penalty of -1, and a gap score of 0, give an efficient sequential algorithm to compute the score of the best local alignment between S_1 and S_2. What is the asymptotic complexity of your algorithm? What are the space requirements? Suppose now that you are given a multi-core processor with p cores (with $1 < p < \min(n, m)$), design and analyze a multicore algorithm for sequence similarity problem using local alignments that scales with the number of cores.

2. A social network graph includes n vertices representing people and m edges, where an edge (i, j) connects v_i and v_j $(i \neq j)$ when person i and person j are friends. A clique is defined as a set of friends who all know each other (that is, there is an edge between each pair in the clique.)

   (a) Give an algorithm to find the largest clique in this social network graph, and prove its complexity.

   (b) The *global clustering coefficient* is a measure of social network connectedness, and is defined as the number of closed triples (which is equivalent to 3× the number of triangles in the graph) divided by the number of connected triples in the graph, where a connected triple is defined as a set of three vertices that are connected by two or three edges. Give an efficient algorithm for computing the global clustering coefficient of a social network graph and analyze its running time.

(c) Give two strategies for parallelizing the computation of the global clustering coefficient on two platforms: a shared-memory multi-core system, and a distributed-memory cluster system.

3. A *cut* in a graph is the set of edges that, when removed, separates a graph into two or more components. A *minimum edge cut* is the smallest number of edges that separates the graph into two components of nearly equal size (i.e., the number of vertices in each component).

   (a) Give an algorithm that can give an approximate minimum edge cut partitioning of the graph. Analyze the cost of the algorithm.

   (b) Give a qualitative analysis of how this method will perform on the following types of graphs:

   - Erdős-Renyi random graphs
   - 3-dimensional torus graph
   - 2-dimensional finite-elements mesh
   - A social network graph (e.g. the Facebook graph)

4. Given n points in 2-d space, give a cache-oblivious data structure that can return nearest neighbor queries.