CSE Qualifying Examination
High Performance Computing
Fall 2022

Instructions: Please answer three of the following four questions. All questions are graded on a scale of 10. If you answer all four, all answers will be graded and the three lowest scores will be used in computing your total.

Questions:

1. A simple performance model for communication between distributed memory processing nodes models communication time as the sum of a latency term and a rate term,

$$t(m) = \alpha + m\beta$$

where m is the message length.

(a) Describe limitations of this performance model on modern compute nodes containing multiple cores per node but only one network interface (NIC) per node, where bottlenecks may occur between the cores and the NIC and between the NIC and the interconnect.

(b) Propose a more suitable performance model for the above situation.

2. Consider the following recurrence, given the real-valued coefficients $a_1, \ldots, a_n$, $b_1, \ldots, b_n$, $c_1, \ldots, c_n$, and $d_1, \ldots, d_n$.

$$x_i = \begin{cases} 0 & \text{if } i = 0 \\ \dfrac{a_i x_{i-1} + b_i}{c_i x_{i-1} + d_i} & \text{if } 1 \le i \le n \end{cases} \tag{1}$$

   (a) Give an efficient parallel algorithm for evaluating this recurrence, analyze it, and argue why it is efficient. (*Hint:* Consider a transformation that allows you to apply a parallel prefix or scan.)

   (b) A well-known method to estimate the square-root of a number, a, is to run n steps of the recurrence $x_i = \frac{x_{i-1}^2 + a}{2x_{i-1}}$, given some initial guess x_0. *(This recurrence comes from Newton's method; note that the numerator contains x_{i-1}* **squared***.)* Suppose we wish to design a parallel algorithm for this problem using only simple addition, subtraction, multiplication, and division. Argue that any algorithm requires $\Omega(n)$ steps, i.e., that no speedup is possible beyond a constant factor.

3. A tensor contraction is a generalized matrix multiplication. Suppose you want to compute

$$C = AB$$

where A and B are 3-dimensional tensors and C is a 4-dimensional tensor. The tensor contraction of interest defines

$$C_{ijkl} = \sum_m A_{ijm} B_{mkl}$$

(a) In the 3-D matrix multiplication algorithm for computing the ordinary matrix multiplication of two matrices, the *computational work* (which scales cubically with the size of the matrices) *is partitioned* among the processors arranged in a 3-D processor mesh. In analogy to the 3-D algorithm, describe a parallel algorithm for computing the above tensor contraction. Be sure to also state how many dimensions is the processor mesh and how the computational work and the tensors are partitioned.

(b) Suppose you are given P distributed memory processing nodes. How should the tensors A and B be partitioned among the nodes in order to reduce communication in the tensor contraction? You do not have to use all P nodes, but assume the tensors are large enough such that *computation* time tends to be smaller if more nodes are used.

4. Consider an undirected graph with n vertices and m edges, where the graph is *sparse* in the sense that $m = O(n)$. A *triangle* in this graph is a set of vertices $\{u, v, w\}$ such that the edges (u, v), (v, w), and (w, u) all exist. Give an algorithm for a distributed-memory machine to count the total number of triangles. Be sure to specify how the input graph is represented and initially distributed. Analyze the communication complexity of your scheme and argue why it should be regarded as "efficient".