HPC Spring 2017 – Homework 4

Robert van Engelen

Due date: April 18, 2017

1. Design a master-worker program to compute the product of two $n \times n$ matrices A and B using a decomposition into $m \times m$ blocks, such that each worker computes the $m \times m$ product of a block. The master is responsible for sending the tasks (two blocks, one for A and one for B) and for receiving the products from the workers and for summing them up into the final result. Determine the parallel t_{comp} and t_{comm} given the constants t_{startup} and t_{data}.

2. Write (pseudo or C) code that implements a parallel Monte Carlo calculation of $\pi/4$ using independent random number streams[1]. See note 16 of *Algorithms PART 1: Embarrassingly Parallel*. Explain how you decided to combine the sub-results per processor to produce the overall estimation of π.

3. Use divide and conquer to compute the n^{th} power of x in parallel by the property that

$$\begin{array}{rcll} x^n & = & x^{n/2} \cdot x^{n/2} & \text{if } n \text{ is even} \\ x^n & = & x^{(n-1)/2} \cdot x^{(n-1)/2} \cdot x & \text{if } n \text{ is odd} \end{array}$$

   Use this algorithm to implement the parallel transitive closure of a undirected graph with n nodes from its $n \times n$ adjacency matrix A where $a_{ij} = a_{ji} = 1$ when nodes i and j are connected. Recall that A^2 is the adjacency matrix that connects nodes at distance 2, A^3 is the adjacency matrix that connects nodes at distance 2 and 3, and so on. Assuming shared memory (no communication cost) and that n is a power of 2, what is the asymptotic parallel time?

[1] You may assume that function `rand()` generates a unique RN as an i.i.d. random variable per processor.