Mastermath midterm examination
Parallel Algorithms

Teacher: Rob H. Bisseling, Utrecht University

October 21, 2020

Each of the three questions is worth 10 points. Total time 120 minutes. Motivate you answers!

1. Let A and B be square $n \times n$ matrices. Assume that we have $p = M^2$ processors, and that $n \bmod M = 0$. Assume that A and B are both distributed by the square block distribution.

   (a) [5 pt] Give an efficient BSP algorithm with 2 supersteps, in the notation we have learned for processor $P(s,t)$, $0 \le s,t < M$, for the computation of the matrix $C = AB$, where on output the matrix C is also in the square block distribution.

   (b) [3 pt] Analyse the BSP cost of your algorithm.

   (c) [2 pt] How much memory does your algorithm need per processor? What would you do if less memory were available?

2. Assume that we have p processors. Processor $P(s)$ has a vector $\mathbf{x}_s$ of length k, for $0 \le s < p$. Assume that $k \gg p$ and that $k \bmod p = 0$. We denote the components of the vector $\mathbf{x}_s$ by $x_s[i]$, for $0 \le i < k$.

   The *reduction* of these vectors computes a vector $\mathbf{x}$ of length k with components $x[i]$, defined by

   $$x[i] = \sum_{t=0}^{p-1} x_t[i], \quad \text{for } 0 \le i < k.$$

   On output, each processor $P(s)$ should have a copy of $\mathbf{x}$.

   (a) [5 pt] Give an efficient BSP algorithm in the notation we have learned for processor $P(s)$ for the reduction. Keep computation and communication cost to a minimum.

(b) [5 pt] Analyse the BSP cost of your algorithm.

3. Let $r > 1$ be an integer, called the *radix*. Let $\mathbf{x}$ be a vector of integers of length n, with $0 \leq x_i < r$ for $0 \leq i < n$. The vector can be identified with a very large integer x, defined by

$$x = \sum_{i=0}^{n-1} x_i r^i.$$

An example would be the choice $r = 10$, where x_i is the ith decimal digit, with x_0 the *least significant digit*.

(a) [5 pt] Assume that we have two input vectors $\mathbf{x}, \mathbf{y}$, which are both distributed by the block distribution over p processors with $n \bmod p = 0$, and which represent very large integers x and y. Assume that all processors know r.

Design an efficient BSP algorithm for the computation of the sum $z = x + y$, You can describe the algorithms in words (instead of the full notation we have learned). Here, z must be stored as a vector $\mathbf{z}$ in the same format as $\mathbf{x}$ and $\mathbf{y}$. In particular, make sure that $0 \leq z_i < r$ by *carrying* a digit to z_{i+1} if needed. You may assume that $x, y < r^n/2$ so that z will fit in the required format.

(b) [5 pt] Analyse the BSP cost of your algorithm. What is the worst case? What do you expect for a random case, where the input digits are randomly chosen integers in the range $[0, r - 1]$?