FDA125 Advanced Parallel Programming February 2007

Exercises for Topic II (PRAM Model and Theory)

Exercise II.1

Give high-level CREW and EREW PRAM algorithms for copying the value of memory location $M[1]$ to memory locations $M[2],...,M[n+1]$.

Exercise II.2

On a RAM the maximum element in an array of n real numbers can be found in $O(n)$ time. We assume for simplicity that all n elements are pairwise different.

(a) Give an EREW PRAM algorithm that finds the maximum element in time $\Theta(\log n)$. How many processors do you need at least to achieve this time bound?

What is the work and the cost of this algorithm? Is this algorithm cost-effective with n processors? With $n/\log n$ processors?

(b) Give an algorithm for a Common CRCW PRAM with n^2 processors that computes the maximum element in constant time.

(*Hint:* Arrange the processors conceptually as a $n \times n$ grid to compare all n^2 comparisons of pairs of elements simultaneously. An element that is smaller in such a comparison cannot be the maximum. Use the concurrent write feature to update the entries in an auxiliary boolean array m of size n appropriately, such that finally holds $m[i] = 1$ iff array element i is the maximum element. Given m, the maximum location i can be determined in parallel using n processors.)

What is the work and the cost of this algorithm? Is this algorithm cost-effective?

Further reading on the maximum problem:

Any CREW PRAM algorithm for the maximum of n elements takes $\Omega(\log n)$ time.
See [Cook/Dwork/Reischuk SIAM J. Comput. 1986]

There exist CRCW PRAM algorithms for n processors that take $O(\log\log n)$ time.
See [Valiant SIAM J. Comput. 1975, Shiloach/Vishkin J. Algorithms 1981]

Exercise II.3

Show that the cost of a cost-optimal parallel algorithm A is of the same order of magnitude as the work of the optimal sequential algorithm S: $c_A(n) = \Theta(t_S(n))$.

Exercise II.4

Give a $O(\log n)$ time algorithm for computing parallel prefix sums on a parallel list. (Hint: Use the pointer doubling technique.)