Parallel Algorithms and Parallel Computers

(in4026)

Examination August 25, 2009

9.00-12.00 hrs

This exam consists of two parts: Part I (architecture) and Part II (algorithms). If you have done one of the two parts before in 2009, you can do the other one. If you have not done one of the two parts in 2009, you have to do both parts.

Please do note that after the August 2009 re-examination, the individual results for Part I and Part II obtained in 2009 or earlier are no longer valid!

IMPORTANT*: Please use a separate sheet for each question!*

PART I

Question 1 *(25 points)*

(5 points)

a. Consider a Completely Connected network (i.e. all the nodes count) with p processors. Derive expressions for the diameter, the bisection width, the arc connectivity, and the number of links.

(5 points)

b. Given is a multistage interconnection network with p processors. Every switching element in the network consists of a *4* by *4* crossbar switch. The interconnection network has 3 stages. How many processors can be connected to the multistage network to make an arbitrary connection between the processors possible?

(5 points)

c. Consider the class of Mesh networks and the class of Hypercube networks. Given are the following three properties:

I. The interconnection costs are proportional to the number of processors.
II. Fast algorithms are asymptotically as fast as the fastest PRAM algorithms.
III. The interconnection network has an elegant recursive structure.

Which property belongs to what class of networks. Explain your answer.

(10 points)

d. The perfect shuffle interconnection pattern can be used in a multi-stage interconnection network for p processors.
  - Derive a mathematical relation between input i and output j of a generic p processor perfect shuffle pattern (Both the number of inputs as the number of outputs are a power of 2)
  - Draw such a multi-stage network interconnection network using this perfect shuffle for 8 inputs and 8 outputs.
  - Show how an automatic route can be established between an arbitrarily chosen input and output in this network.

Question 2 *(25 points)*

(5 points)

a. Given is a d-dimensional hypercube with $p = 2^d$ processors. What is the maximal communication time of a message of length m if we use *Store-and-Forward* routing.

(5 points)

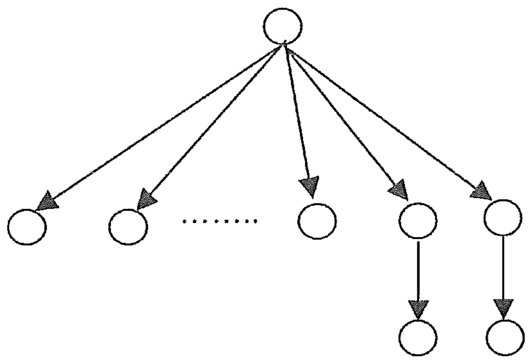
b. Given is a p processor 3D mesh with wrap around. What is the communication time of a message of length m for a *one-to-all* broadcast using *Store-and-Forward* routing.

(5 points)

c. Give the definition of the overhead function T_o in terms of the amount of work (W) and the processor-time product. Derive this function for the addition of n numbers on a p processor Hypercube. Given is that both one addition and a single communication step to (an arbitrary) other processor take one unit time.

(10 points)

d. Given a parallel processor consistent of p processors. Given is also the following directed acyclic task graph (DAG) with $N=2^n-1$ tasks ($n>1$).



The graph consists of tasks that all have unit execution times. The nodes of the graph are mapped (by some scheduler) onto the parallel processor (we assume that every processor has at least one task assigned). The communication times between the processors can be neglected.

1. How long is the critical path of the graph?
2. What is the maximal degree of concurrency in the graph?
3. Give speed up, efficiency, and overhead functions in case of maximal concurrency?
4. What is the minimum number of processors to have the parallel execution time T_p equal to the critical path? Is this number Cost-Optimal?

Part II

Question 3 *(25 points)*

Let $A[1..n]$ and $B[1..n]$ be two arrays of integers.

a. *(5 points)* Give a $T(n) = O(1)$ and $W(n) = O(n)$ algorithm on a CRCW that, given A and B, returns *true* if $A = B$ and returns *false* else.

b. *(10 points)* Give a $T(n) = O(1)$ and $W(n) = O(n^2)$ algorithm on a CRCW to decide whether or not for every element $A[i]$ of A there is an element $B[j]$ of B such that $A[i] = B[j]$.

c. *(10 points)* Assume that all elements $A[i]$ of A are unique, i.e., for all $1 \leq i, j \leq n$, $i \neq j$ implies $A[i] \neq A[j]$. Give a fast weakly optimal parallel algorithm on a CREW that checks whether the array B is a permutation of the array A.

-relate to b -faster than obvious non-w.o. using b

Question 4 *(15 points)*

Consider the following (third-order) linear recurrence

$$y_0 = 1$$
$$y_1 = 2$$
$$y_2 = 2$$
$$y_i = y_{i-1} + 2y_{i-2} + 2y_{i-3} \text{ for } i = 3, \ldots, n-1.$$

Let $\mathbf{y} = [y_0, \ldots, y_{n-1}]$.

a. *(5 points)* Show how this linear recurrence can be reduced to a first-order linear recurrence.
b. *(5 points)* Show that there exists a parallel algorithm to compute $\mathbf{y}$ in time $T(n) = O(\log n)$.
c. *(5 points)* What is the work-complexity of your algorithm.

Question 5 *(10 points)*

Let L be a non-empty list of n elements. Every element y of L has a pointer $p(y)$ pointing to the next element in the list L or to nil if y is the last element of L. Develop a fast parallel algorithm that for every element x in the list L calculates the number of elements in L that come after x (You may assume that computing the sum of two integers can be done in constant time).
What is the time and work complexity of your algorithm?

End of Examination