UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2003

MEng Honours Degree in Information Systems Engineering Part IV
MSci Honours Degree in Mathematics and Computer Science Part IV
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
PhD
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C429=I4.10

PARALLEL ALGORITHMS

Thursday 8 May 2003, 14:30
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators required

1. a Describe the principles of operation of the *Parallel Random Access Machine* (PRAM) idealised model of computation. Explain the differences between the exclusive and concurrent modes of memory access.

b i) Define the terms speedup (S_p), efficiency, (E_p) and cost (C_p) relating to the run-time (T_p) of an algorithm executing on a p-processor parallel architecture;

ii) What is a *cost-optimal* algorithm? Explain your answer in terms of both efficiency and cost. How can increasing the *grain size* of an algorithm make it more likely to be cost optimal?

iii) What is meant by the term *scalability*? Define the *isoefficiency function* and explain in what sense it is a metric for scalability.

iv) Can an architecture with constant communication times be non-cost-optimal? Give an example to explain your answer.

c i) Write an algorithm that performs multiplication of an n by n matrix and a n-component vector on a PRAM of n^2 processors in log n time, each processor performing just one scalar multiplication.

ii) Is your algorithm cost-optimal on a CREW PRAM? Is it cost-optimal on any type of CRCW PRAM?

The three parts carry, respectively, 20%, 50% *and* 30% *of the marks.*

2. a i) Give a *non-recursive* parallel algorithm to transpose an $n \times n$ matrix $A = (a_{ij} \mid 1 \le i,j \le n)$, giving the result matrix A' with components $a'_{ij} = a_{ji}$ $(1 \le i,j \le n)$, on a *d-dimensional hypercube* architecture, where $n^2 = 2^d$. What is its parallel run time?

ii) Using the property that the transpose of a matrix partitioned into block matrices is the block-transpose of the transposed blocks, give a *recursive* algorithm to transpose the same matrix on the same hypercube. Compare the run-time with your algorithm of part a i).

b i) For a function $u(x,y)$ of two variables, give the discrete approximations for the first derivatives $\partial u/\partial x$, $\partial u/\partial y$ and the second derivatives $\partial^2 u/\partial x^2$, $\partial^2 u/\partial y^2$ using *central differencing*, i.e. using a NEWS grid.

ii) Compare and contrast the solution methods for the following partial differential equations defined in a finite region $\mathcal{R}$ in 2-dimensional space:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y); \quad y\frac{\partial^2 u}{\partial x^2} + x\frac{\partial^2 u}{\partial y^2} = f(x,y)u; \quad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y)u^2; \quad \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = f(x, y)$$

for a given function $f(x,y)$ with u given on the boundary of $\mathcal{R}$.

iii) For the first of these equations, suppose there is a processor for each point in your grid. Describe a parallel algorithm for the solution of the resulting equations. How would your algorithm change if a square array of points were allocated to each processor?

The two parts carry the same weight.

3. Consider a one-to-all personalized broadcast that takes place on a p-processor hypercube with bi-directional links and store-and-forward routing. A single source processor starts with a vector of p unique messages $M=\{m_0, m_1, \cdots, m_{p-1}\}$, where message m_i is destined for the processor numbered i. Every message m_i is of length l. An algorithm for performing this operation is:

```
procedure one_to_all_personalized(node, d, M)
  result = M
  for i = d-1 downto 0 do
    partner = node XOR 2^i
    if (is_sender(node, i)) then
      /* send second half of result to partner */
      transmit = subvector(result, 2^i, 2^(i+1) -1)
      send(partner, transmit)
      /* retain first half of result on sender */
      result = subvector(result, 0, 2^i -1)
    else if (is_sender(partner, i)) then
      receive(partner, result)
    end if
  end for
end procedure
```

This program is executed by each processor in the hypercube. Here `node` is the processor's number/label and `d` is the dimension of the hypercube ($p = 2^d$). If `node` is zero (i.e. the source), `M` is the vector of p messages (each of length l) to be broadcast; otherwise `M` is empty. `XOR` is the bitwise exclusive-or operator and `is_sender`(n, i) is true if the $i+1$ least significant (rightmost) bits of the binary representation of n are all zero. Given a vector $V = \{v_0, v_1, \cdots, v_{|V|-1}\}$, the function `subvector`($V$, *start*, *end*) returns the sub-vector $\{v_{start}, v_{start+1}, \cdots, v_{end}\}$.

a) Given an 8 processor hypercube and source vector M={A, B, C, D, E, F, G, H} (here p=8 and l=1), draw a diagram showing the communication that takes place and the contents of the `result` vector on each hypercube node for each step of this algorithm.

b) Derive a *general* expression for the parallel run time of this algorithm. You may assume start-up time is t_s, hop time is negligible, per-word transfer time is t_w and the time taken to construct a subvector with k l-word elements is $kl\,t_a$.

c) Derive the run-time of a sequential (1 processor) implementation which constructs p vectors each containing a single l-word message from a source vector M (where M contains p l-word messages). Is the parallel algorithm cost optimal?

d) Predict the maximum speedup and corresponding efficiency of the algorithm that can be achieved (for messages of any length) when broadcasting personalized messages on an 8 processor hypercube with t_s = 100ns, t_w = 60ns and t_a = 120ns.

e) To which MPI primitive does this operation correspond?

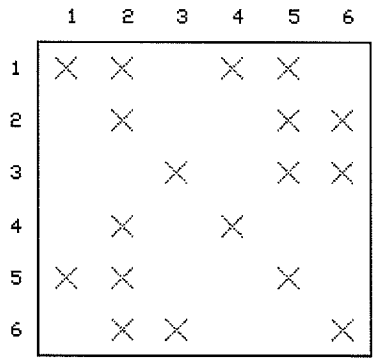
The five parts carry, respectively, 25%, 25%, 20%, 20% and 10% of the marks.

4.a) State whether each of the following statements is True or False. In each case, give a *brief* (one sentence) justification for your answer.

i) An algorithm designed for a CRCW PRAM with an *arbitrary* write arbitration protocol will exhibit similar complexity when executed on CRCW PRAM with a *priority* write arbitration protocol.

ii) Consider a parallel algorithm with isoefficiency function $W = p\log_2 p$, where W is the problem size and p is the number of processors. To maintain a constant efficiency when p is doubled, W must increase by a factor of $2(1+\log_2 p)$.

b) Consider the task of performing a row-striped parallel sparse-matrix vector product over two processors, given that the sparse matrix has the following non-zero structure:



i) Under a naïve row-striping approach, what is the total communication cost per sparse-matrix vector product?

ii) Define a hypergraph $H = (V, N)$ that represents this sparse matrix.

iii) Draw the hypergraph, and partition it into two *balanced* partitions such that the hyperedge cut is minimized.

iv) Using your partitioning, show how the rows and columns of the sparse matrix (and corresponding vector elements) may be permuted such that the total communication cost per sparse-matrix vector product is minimized. What is the total communication cost now?

The two parts carry, respectively, 20% and 80% (10%+15%+30%+25%) of the marks.

End of paper