UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1996

MEng Honours Degrees in Computing Part IV
MSc Degree in Foundations of Advanced Information Technology
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the
Diploma of Membership of Imperial College
Associateship of the City and Guilds of London Institute

PAPER 4.29

PARALLEL ALGORITHMS

Tuesday, April 30th 1996, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains
4 questions
3 pages (excluding cover page)

1 For this question you may assume that each communication incurs a start-up time, t_s, a per-hop header transmission time, t_h, and a per-word data transmission time t_w.

a Explain the principles of operation of the *store-and-forward* routing and *cut-through* routing schemes and explain why cut-through routing yields lower latency. As part of your answer derive expressions for the maximum latency of message transmission in a p-processor ring assuming both store-and-forward and cut-through schemes.

b For both a $\sqrt{p} \times \sqrt{p}$ wraparound mesh and a p-processor hypercube write down expressions for:

i The total number of links in the system

ii The average number of links between two nodes

iii The average latency of each communication assuming cut-through routing

c A typical measure of the *cost* of a parallel computer is the total number of *wires* it takes to build it, where each link between two processors is assumed to be constructed from a collection of m>0 wires. The communication *cost-performance* measure of a parallel computer is the ratio of the cost of the computer (in wires) to its performance (in terms of average communication latency between two processors). The bandwidth per link is assumed to scale linearly with the number of wires it contains.

i If a p-processor hypercube contains one wire per link (m=1) how many wires per link will there be in a $\sqrt{p} \times \sqrt{p}$ wraparound mesh with the same cost? Write down the expressions for the cost-performance of each.

ii For what value of p is the cost-performance ratio of a wrap-around mesh the same as that of a hypercube with the same cost? Compare the cost-performance characteristics of both machines as the number of processors is increased beyond this point, assuming that the data transmission time is the dominant factor in each communication.

2a Compare and contrast the alternative data partitioning techniques for matrix operations *block striping (by row or column)* and *block checkerboarding.* Give an example of an algorithm in which you would expect *cyclic striping* to provide a more efficient implementation.

2b Consider the inner-product of the $n \times n$ matrix $\boldsymbol{M}$ and the vector $\boldsymbol{v}$ on a network of n processors, giving the result vector $\boldsymbol{w}$ with components $w_i = \sum_{j=1}^{n} m_{ij} v_j$, $1 \leq i \leq n$.

i Describe two parallel algorithms to execute the product on n processors using first *row-striping* and then *column-striping* data partitioning schemes.

ii Briefly outline how to modify the *row-striping* algorithm to implement *block-striping* in the event that there are fewer than n processors.

iii Now suppose that n is a perfect square. Describe an algorithm for the product that uses a *block-checkerboarding* partitioning.

iv Compare the performance of the algorithms of parts i) and iii) in terms of parallel execution time, but *excluding* communication time.

The two parts carry, respectively, 40% *and* 60% *of the marks.*

3 The "Upper Glass" glazing company is a specialist supplier of smash-resistant glass for shop windows. All glass sold is cut from large "master" sheets and currently the cutting is done on demand and is controlled manually. It has been observed, however, that a lot of glass is wasted because the cutter has no way of judging where to cut the current master sheet to reduce long-term waste. A sequential optimisation program has therefore been written which will process the stream of required cuts C_1, C_2, ..., *in that order*, and work out where to position each cut on the master sheet in order to minimise the amount of wasted glass. Each *cut* specifies the dimensions of the required sheet of glass. The program employs a *breadth-first branch-and-bound* search strategy which terminates immediately if a solution is found with zero waste and which terminates a given subsearch if the unrecoverable waste glass that can be expected is greater than that of the best solution found so far. It has been established that the algorithm is the fastest sequential algorithm for solving the problem.

a Unfortunately the execution times of the sequential program are too long to make it usable and it has been decided to implement a parallel version of the program. The following procedures are already defined:

`Options( C : CutSpec, M : Master ) : [ Position ]`
Given a required cut `C` and the currently proposed (and incomplete) master sheet layout `M` returns a list of all the positions on `M` where `C` can be placed

`Waste( Cs : [ CutSpec ], M : Master ) : real`
Given a list of cuts `Cs` *which have yet to be made* and the current master sheet layout `M` returns the minimum waste area which is inevitable by any allocation of `Cs`

`Spare( M : Master )`
Returns the total unallocated area in `M`

`Add( C : CutSpec, M : Master , P : Position )`
Adds cut `C` to master sheet `M` at the specified position `P`

Explain *briefly* how the parallel version will work. As part of your answer outline the structure of the main procedure

```
BFB&B( Cs : [ CutSpec ], M : Master )
```

for controlling the search for the optimal layout of *one* master sheet, `M`, given the list of required cuts `Cs` (whose total area you may assume to be at least that of a master sheet). *Avoid all unnecessary detail – you are not required to produce a working program.*

b The parallel program is now executed with various input streams and the execution times compared with those from the sequential version. The following were observed when the program was run on p processors:

i In most cases the speedup over the single processor executing the same algorithm was found to be *less* than p.

ii In some cases the speedup over the single processor executing the same algorithm was *greater* than p.

iii In one experiment the branch and bound optimisation was turned off (by commenting out bits of the code) yet the program ran no slower than before.

Suggest what might be happening in each case to account for the behaviour.

The two parts carry, respectively, 70% and 30% of the marks.

Turnover

4a Compare and contrast *qualitatively* the *finite differencing* and *finite elements* methods for solving second-order partial differential equations.

4b Consider a mesh of points labelled $\{(i,j) \mid 0 \le i,j \le n\}$, where point (i,j) represents the coordinate (x_i, y_j) and $x_{i+1} - x_i = y_{i+1} - y_i = h$ for $0 \le i \le n-1$. The first derivative $u_x \equiv \partial u / \partial x$ with respect to x of a function $u(x,y)$ at the point midway between coordinates represented by (i,j) and $(i+1,j)$ is approximated by $(u_{i+1,j} - u_{ij})/h$ where $u_{ij} \equiv u(x_i, y_j)$; $u_y(x_i, y_j + h/2)$ is approximated similarly.

i Define a five-point NEWS grid centred on the point (x_i, y_j), $0 < i,j < n$.

ii Give approximations for the derivatives u_x and u_y at (x_i, y_j).

iii Give approximations for the second derivative $u_{xy} \equiv \partial u_y / \partial x$ at (x_i, y_j) by applying your result for part ii) to the function u_y, itself approximated by part ii). At which points outside your NEWS grid do you require values for u?

4c The partial differential equation $u_{xy} = 0$ with initial conditions $u(x,0) = 1/(1+x)$ and $u(0,y) = 1$ is defined for $x \ge 0$, $y \ge 0$. Derive a set of linear equations that approximately solve this equation in the finite square region bounded by the coordinate axes and lines $x=1$, $y=1$, using a square mesh of size $h/2$. What scope for parallel computation of the solution is there?

End of paper

