UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1997

MEng Honours Degrees in Computing Part IV
MEng Honours Degree in Information Systems Engineering Part IV
MSc Degree in Advanced Computing
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 / I4.10

PARALLEL ALGORITHMS

Thursday, April 24th 1997, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains 4 questions

1 a Describe the *Parallel Random Access Machine* (PRAM) idealised model of computation. Under what circumstances does a real parallel computer have the same *asymptotic* run times?

b i) Describe an algorithm for multiplying two $n \times n$ matrices using n^3 identical processors that executes in $\Theta(\log n)$ time on a PRAM.

ii) What would be the parallel run time on a hypercube interconnection of the same n^3 processors? You may assume n is a power of 2.

iii) Suppose now that you have only p processors where p divides n^3. Explain how your algorithm generalises using a block checkerboarding data partitioning scheme.

2 a i) Explain the data partitioning scheme *block striping by row* for matrix operations.

ii) Consider the transpose operation on an $n \times n$ matrix $\boldsymbol{M}$, giving the result matrix $\boldsymbol{M'}$ with components $M'_{ij} = M_{ji}$ $(1 \le i,j \le n)$. Describe a parallel algorithm to implement the transposition on a *mesh* of p processors, where p divides n, using a block striping by row data partitioning scheme.

iii) Determine the parallel run time of the algorithm of part a ii) executing on a hypercube architecture, given that the number of processors is a power of 2. You may neglect startup and per-hop times, i.e. only consider per-word transfer times, and approximate the number of exchanges needed to transpose an $n \times n$ matrix by $n^2/2$. Is the algorithm cost-optimal on this architecture, and why?

b Using the property that the transpose of a matrix partitioned into block matrices is the block-transpose of the transposed blocks, describe a recursive algorithm to transpose an $n \times n$ matrix on a hypercube of p processors, where n and p are both powers of 2 with $p \le n$.

Turn over/...

3 a Explain how a partial differential equation may be transformed into a set of linear equations by *finite differencing*. When is Gaussian elimination (i) inappropriate but possible; (ii) impossible?

b 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.

c Consider the partial differential equations:

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = x + y$$ with boundary condition $u = 5x^2/2$ when $y=0$, and

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = x + y$$ with boundary condition $u=0$ for (x,y) on the unit square with vertices at coordinates (0,0), (0,1), (1,1), (1,0).

Compare and contrast the solution methods of the linear equations obtained through finite differencing. Hint: consider also *simple* differencing.

The three parts carry, respectively, 35%, 25% *and* 40% *of the marks.*

4 a Define the following metrics relating to the performance of an algorithm executing on a parallel computer architecture comprising p processors:

i) Run time, T_p

ii) Speedup, S_p

iii) Efficiency, E_p

iv) Cost, C_p

b i) What is meant by the term *scalability*?

ii) The *isoefficiency* $I(p, E)$ of a parallel system with p processors running an algorithm at efficiency E is the solution (for W) of the equation $W = \frac{E}{1-E} O_p(W)$ where $O_p(x) = C_p - x$ is the *overhead*, i.e. the amount of computation not performed in the best serial algorithm, for problem size x. In what sense does isoefficiency measure the scalability of the system and why?

c A dynamic parallel algorithm distributes work using the global round robin scheduling strategy on an architecture with p processors.

i) Explain how contention for a certain pointer contributes to the overhead of the parallel implementation.

ii) *Assuming* that $O_p(W) = \Theta(T.p\ .\log W)$ where T is the time to distribute one piece of work, that T is dominated by the time spent by a processor waiting for the pointer which is $1/(1-\rho p)$ on average (where ρ is a constant, $\rho p < 1$), show that the isoefficiency of the system is greater than $\Theta(p(1-\rho p)^{-1} \log (1-\rho p)^{-1})$.

End of Paper