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 4.97

PARALLEL DECLARATIVE SYSTEMS
Friday, May 17th 1996, 10.00 - 12.00

Answer THREE questions

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

1a Outline the main issues of co-ordination in parallel programming and explain how these issues are addressed by the SCL structured parallel programming approach.

b Use SCL co-ordination forms to specify a SPMD parallelisation of the following sequential program for execution on an n-processor distributed memory parallel computer:

```
Function computepi (int : interval)
    \* approximation to pi *\
    int i;
    double sum, width, x;
    sum = 0.0;
    width = 1.0 / interval;
    for i =1 to interval step 1 do {
        x = ( i - 0.5 )*width;
        sum = sum + (4.0/(1.0 + x*x));
    }
```

c Discuss how you would model the performance of your program.

The three parts carry, respectively, 30%, 40%, 30% of the marks.

2a Discuss the major differences between the data distribution directives of High Performance Fortran, Linda tuple spaces and SCL co-ordination forms.

b Recursive bisection takes a divide-and-conquer approach to partitioning a domain into subdomains of approximately equal computational cost. The domain is first cut in one dimension to yield two subdomains. Cuts are then made recursively in the new subdomains. Given a grid represented as a two dimensional array, define a partition function in SCL to partition the grid into n row blocks based on the recursive bisection method. A function *f* can be assumed to measure the load of an array block .

c The Jacobi method is an iterative algorithm to solve linear equations of the form Ax=b. It use values computed for each variable x_i during iteration t to generate new values for iteration t+1:

$$x_i\ (t+1) = 1/a_{ii}\ (b_i - \Sigma_{(j \neq i)}\ a_{ij}\ {*}x_i\ (t)\ )$$

This computation can be implemented in sequential code as

```
repeat
 do  I = 1, N
    x[i]  = b[i]/a[i,i]
 enddo

 diff = 0
 do i = 1, n
 newx[i] = b[i]
         do j =1, n
               if j not.eq. i then
                      newx[i] = newx[i] - a[i,j]* x[j]
               endif
        enddo
 newx[i] = newx[i]/a[i,i]
 enddo

 do i=1,n
        diff = max(diff, abs ( x[i] -newx[i] ) )
        x[i] = newx[i]
 enddo
 until diff < epsilon
```

Parallelise the above sequential code into a SPMD program using the co-ordination forms of SCL.

The three parts carry, respectively, 30%, 40% and 30% of the marks.

Turn over ....

3a Discuss the three general forms of composition: sequential, parallel and concurrent and their co-ordination representations.

b Master_and_Slave (MS) parallel algorithms solve a problem by dividing the problem into a number of independent computations and then combining the results of these computations. Write co-ordination forms to realise two versions of MS : one based on sequential composition where the slave tasks are the same and the other based on parallel composition where the slave tasks can be different.

c Discuss a possible implementation approach for the SPMD and MPMD co-ordination forms in MPI.

The three parts carry, respectively, 25%, 30%, and 45% of the marks.

4a Discuss the co-ordination issues of adaptive computation. Give a practical example of parallel adaptive computation.

b Discuss the relationship between Divide-and-Conquer algorithms and the adaptive domain decomposition model. Revise the Divide-and-Conquer skeleton to capture the additional adaptive computation feature.

c Using the revised Divide-and-Conquer skeleton write the force calculation code Treeforce(i, t) used in the Barnes-Hut algorithm to compute the force on a particle i of all the particles organised in a Quadtree t. You may assume that any sequential program, such as a program for computing the gravitational function, required to perform local computation is already defined.

The three parts carry, respectively, 25%, 30%, 45% of the marks.

End of Paper