UNIVERSITI SAINS MALAYSIA

First Semester Examination
Academic Session 1999/2000

September 1999

CSC503 - Foundations of Parallel Computing

Duration : [3 hours]

INSTRUCTION TO CANDIDATE:

- Please ensure that this examination paper contains **FOUR** questions in **FIVE** printed pages before you start the examination.
- Answer **ALL** questions.
- You can choose to answer either in Bahasa Malaysia or English.

ENGLISH VERSION OF THE QUESTION PAPER

1. (a) Assuming that you are a parallel expert, what advice would you give for each of the following cases. Justify your suggestions.

   (i) Case I
   The head of the Meteorology Department is considering whether or not he should parallelise the existing sequential climate modeling. The current program takes 1.5 days to run and he is expecting to use a more complex model to get a more accurate prediction.

   (ii) Case II
   A PhD student at the Physics Department-USM is doing research on the behaviour of a system of particles. His current program takes 7 hours to run. He has obtain a dozens set of results and planning to get a dozen more. He is contemplating whether to have his program parallelised.

   (iii) Case III
   An MSc student develops a new rendering algorithms which takes 30 minutes to run. Would you advice him to parallelise his algorithm.

   (9/100)

   (b) Explain the steps taken to do reverse engineering for the above sequential programs. Suggest 3 CASE tool that can be used to assist the conversion of the sequential programs in (a).

   (6/100)

   (c) Explain the advantages of virtual topology and derived datatype and give examples when you would use them.

   (4/100)

   (d) Compare the two dynamic load balancing algorithms discussed in class, namely Diffusion and Dimension Exchange Method. Discuss the suitability of the algorithms on a hypercube, 2-d mesh and network of workstations.

   (6/100)

2. (a) Compare and contrast between shared memory, distributed-memory (message-passing) and SIMD data-parallel programming model.

   (6/100)

   (b) Explain the two scheduling method below:

   (i) Scheduling in-forest/out-forest tasks graphs.

   (ii) Scheduling interval ordered tasks.

   (4/100)

(c)

```
While (not finished) do
      compute ()
      load_balance ()
      barrier ()
endwhile
```

The above is an outline of a Bulk Synchronous Parallel (BSP)-like computational model where computation is divided into a sequence of iterations. Each iteration is delimited by a barrier. Within each iteration processors perform local computation and global communications independently.

(i) Develop a general performance model for the total parallel execution time $\mathbf{T_{par}}$. Use the following parameters for your model:

$\mathbf{T_{cp}}$ - the time each procesor spends on local computation.
$\mathbf{T_{lb}}$ - the time each processor execute the load balancing operation.
$\mathbf{T_{sy}}$ - the time for the barrier.
$\mathbf{I}$ - the maximum number of iterations.

For simplicity, assume there is no idle time incurred in the above computation.

(ii) Determine the cost model of each term, that is $\mathbf{T_{cp}}$, $\mathbf{T_{lb}}$ and $\mathbf{T_{sy}}$.

The barrier operation takes a simple logarithmic time, a function of the number of processors. Each processor communicates with two of its neighbouring processors exchanging 4 byte data at a time. The local computation executes on a 2-d grid of size X by Y and each grid points take $\mathbf{t_{avg}}$ to compute.

(iii) Is the above computational model suitable for network of workstations? Explain your answer.

(15/100)

3. (a) (i) Distinguish between static and dynamic interconnection network (IN).

(ii) Describe the type of applications suitable for each static and dynamic IN.

(5/100)

(b) List 3 factors that contribute to overhead (and thus limit speedup) in parallel programs.

(3/100)

(c) It is possible to construct a system that is a hybrid of a message passing multicomputer and a shared memory multiprocessor.

(i) Describe how this can be achieved.

(ii) List it's relative advantages over a purely message passing system and a purely shared memory multiprocessor system.

(3/100)

(d) Differentiate between Amdahl's Law and Gustafson's Law based on the assumption that they made in deriving the laws.

(2/100)

(e) Given a CREW PRAM algorithm for rank sort which sorts numbers by counting the number of numbers less than each number and using this count to reposition the number under consideration.

```
forall Pi where 0 ≤ i < n do
      x=0
      for j= 0 to n do
            if a[i] > a[j] then x <- x+1 endif
      b[x] <- a[i]
      end for
end for
```

(i) Modify the above algorithm for CRCW PRAM.

(ii) What is the time complexity of the algorithm in (i)?

(4/100)

(f) Scatter and gather are classified as collective communication routines.

(i) Describe the scatter and gather operations.

(ii) If scatter and gather routines were not provided by the message passing system, write separate pesudocode to implement the scatter and gather operation.

(8/100)

4. Suppose a sequence of numbers $x_0, x_1, ..., x_{n-1}$ are to be added up. Divide the sequence of numbers into m parts of n/m numbers each, at which m processors (or processes) can each add one sequence independently to create partial sums. The m partial sums need to be added together to form the final sum.

Your task is to design a message passing program using the master-slave programming model for the above problem.

(a) Draw a diagram which illustrates the master-slave model for the above problem, indicating clearly in terms of n and m, the different subsequence assigned to the processes. Consider a general case of m processes.

(3/100)

(b) Write PVM-like pseudocode for the above problem using send() and receive() statements to transfer data between master and slave.

(5/100)

(c) If a broadcast/multicast routine is used to send the complete list to every slave, code is needed in each slave to select that part of the sequence to be used by the slave. Again, write PVM-like pseudocode to accomplish this.

(5/100)

(d) Write PVM-like pseudocode to solve the same problem, this time using the scatter and gather routine. (5/100)

(e) Discuss the weaknesses and strengths of each implementation in (b), (c) and (d). (4/100)

(f) In your opinion, which implementation is likely to give the best performance? Support your answer with a clear explanation. (3/100)

- oooOOooo -