NATIONAL UNIVERSITY OF SINGAPORE

CS3211 - PARALLEL AND CONCURRENT PROGRAMMING

(Semester 2: AY2015/2016)

Time allowed: 2 hours

INSTRUCTIONS TO STUDENTS

This assessment paper contains **FIVE (5)** sections totalling **FORTY (40)** marks, and comprises **FOURTEEN (14)** printed pages including this one.

This is an **OPEN BOOK** assessment, and you are to answer **ALL** questions. You may cite any result in the lecture notes or tutorials. Answer **ALL** questions within the space provided in this booklet (write on the backs of pages if you need more room).

Please write your Student Number below.

STUDENT NO: ______________________________

This portion is for examiners use only.

Question		**Marks**	**Remark**
General topics, short answers	Q1 (10)		
Speedup and analysis	Q2 (12)		
Accuracy and algorithms	Q3 (4)		
Programming	Q4 (7)		
Models	Q5 (7)		
Total:	Q1-5 (40)		

Q1 (Short Answer Questions) (10 marks)

In the following 6 short questions, write a brief answer in the box provided. Each answer is worth 1 (ONE) mark.

1.1 Classify the Micron Automata processor system in the Flynn taxonomy as SISD, SIMD, MISD, or MIMD, and give a one-sentence explanation for your choice.

Answer:

1.2 Classify the "Tembusu" system you used in your projects in the Flynn taxonomy as SISD, SIMD, MISD, or MIMD, and give a one-sentence explanation for your choice.

Answer:

1.3 Classify a Graphics Processing Unit (GPU) made in the year 2005 in the Flynn taxonomy as SISD, SIMD, MISD, or MIMD, and give a one-sentence explanation for your choice.

Answer:

1.4 Classify a Turing machine in the Flynn taxonomy as SISD, SIMD, MISD, or MIMD, and give a one-sentence explanation for your choice.

Answer:

Q1 (Short Answer Questions) (Continued)

1.5 Differentiate between an NFA and a DFA. What property of the states of an NFA is of importance in parallelism?

Answer:

1.6 CSP#/PAT have assertions that can differentiate between *nonterminating*, and *deadlockfree* processes. Explain the difference between *nontermination*, and *deadlockfreeness*.

Answer:

1.7 Explain how a semaphore may be used to ensure processes are mutually excluded from a section of code. Briefly outline how you would use the semaphore in each process, and what its initial value would be.

Answer:

Q1 (Short Answer Questions) (Continued)

1.8 One of the necessary and sufficient conditions for deadlock is *Incremental acquisition*: processes hold on to resources. Briefly give a situation where two processes can deadlock while trying to get exclusive access to two files. Your brief example should rely on the incremental acquisition condition.

Answer:

1.9 Briefly differentiate between a *monitor*, and a *semaphore*.

Answer:

1.10 Certain concrete features of MPI are related to more abstract features in CSP. An example might be CSP shared events. What is the corresponding concrete feature of MPI?

Answer:

Q2 (Speedup and analysis) (12 marks)

There is a technique for finding the optimal filter for a blurry N-by-N image that takes time proportional to the fourth power of N, after some startup time. Specifically, the time required for a problem of size N running on P processors is

$$t(N,P) = 0.5 + 10^{-9}\frac{N^4}{P}\text{seconds}$$

2.1 Find the fixed-size problem speedup for $N = 256$ and $P = 1$, 16, or 256. (1 mark)

Answer:

2.2 Find a formula for N as a function of P that keeps $t(N,P)$ equal to what it is for $t(256,1)$, the serial time. (1 mark)

Answer:

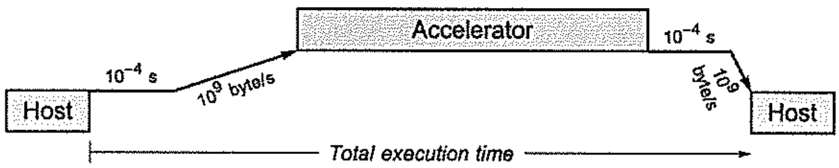
2.3 Use the answer from 2.2 to find the fixed-time problem speedup for $P = 1$, 16, 256. (2 marks)

Answer:

Q2 (Speedup and analysis) (Continued)

A host processor is attached to an accelerator. The task is one in bioinformatics, to compute how close a match a genetic sequence is to a genome. The genome resides on both the host and the accelerator, but each genetic sequence is 1000 bytes long and must be sent to the accelerator. The result of each match is a 4-byte value that must be returned to the host. The host takes 10^{-5} seconds to compute the match. The accelerator is ten times faster, and can compute a match in 10^{-6} second.

A naïve user applies the accelerator as an "offload engine" to do the work. Data can be sent to or from the accelerator at the rate of 10^9 bytes/second after a startup latency of 10^{-4} seconds, visualized as in the diagram below.



2.4 How long, in seconds, will this approach take to do n matches? (1 mark)

Answer:

2.5 How large does n have to be for this approach to be faster than simply running the problem on the host? (2 marks)

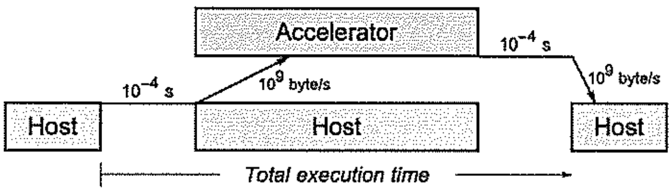
Answer:

Q2 (Speedup and analysis) (Continued)

2.6 As n grows large, what is the asymptotic speedup compared to using only the host?(1 mark)

Answer:

Now assume the accelerator is used more intelligently. The host helps by calculating the forces on $\frac{1}{11}$ of the sequences and the accelerator does the other $\frac{10}{11}$ since it is 10 times faster, so they should take roughly equal amounts of time. Also, the transfer of data to the accelerator is overlapped with computation time, except for the startup.



2.7 Again find the formula for the total execution time. (1 mark)

Answer:

Q2 (Speedup and analysis) (Continued)

2.8 For the approach used in 2.7, find the values of n large enough that the use of the accelerator is beneficial. (2 marks)

Answer:

2.9 For the approach used in 2.7, find the asymptotic speedup for large n compared to using only the host. (1 mark)

Answer:

Q3 (Accuracy and algorithms) (4 marks)

An array of 1024 floating-point numbers has random values ranging from 1.00 to 2.00.

3.1 Describe a strategy for computing their sum that (on average) results in the least amount of cumulative rounding error. (1 mark)

Answer:

3.2 How does this compare with the strategy you would use if the numbers started on a parallel computer with 1024 processors, one number per processor? (1 mark)

Answer:

Q3 (Accuracy and algorithms) (Continued)

You have an array of 23 numbers x_0 to x_{22}, and 5 processors P_0 to P_4. You want to allocate a nearly equal subset of numbers to each processor without leaving anything out and without duplication.

3.3 Find the formula for `j = start(i)`, the x index that processor P_i has as its first element, and `k = length(i)`, the length of the x array on processor P_i. (1 mark)

Answer:

3.4 Fill out the table for j and k as a function of i. (1 mark)

Answer:

Q4 (Programming) (7 marks)

4.1 Cannon's algorithm is a well known algorithm for performing matrix-matrix multiplications. Explain the relevance and use of cartesian topologies in implementing Cannon's algorithm with MPI. (3 marks)

In your answer you should clearly explain the use in MPI of the term "cartesian topology". You should explain how a cartesian topology eases or improves the Cannon code.

Q4 (Programming) (Continued)

In the odd-even bubble sort described in class, the parallel runtime was given as

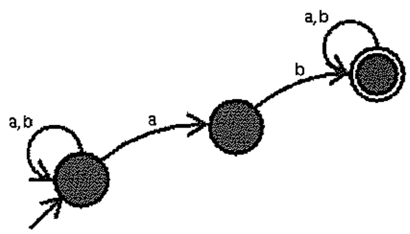
$$T_P = \overbrace{\Theta\left(\frac{n}{p}\log\frac{n}{p}\right)}^{\text{local sort}} + \overbrace{\Theta(n)}^{\text{comparisons}} + \overbrace{\Theta(n)}^{\text{communication}}.$$

4.2 Explain the term $\Theta\left(\frac{n}{p}\log\frac{n}{p}\right)$. Explain why it has this value. (2 marks)

4.3 Explain the communication term in the expression for parallel runtime. On what sort of network is the communication term $\Theta(n)$? Can you do better? (2 marks)

Q5 (Models) (7 marks)

The following NFA describes a recognizer for a particular language:



5.1 Describe (in words) the language accepted by this NFA. (2 marks)

Answer:

5.2 In this NFA, what is the maximum degree of parallelism, expressed as a proportion of active states? Give a string that will lead to this degree of parallelism. (2 marks)

Answer:

Q5 (Models) (Continued)

5.3 Modify the NFA from Q5.1 so it will now also accept the same strings, but interspersed with other (non-a, non-b) characters. Draw your NFA in the box below. (2 marks)

Answer:

5.4 In which part of a CSP# source are LTL expressions found? (1 mark)

Answer:

=== END OF PAPER ===