NATIONAL UNIVERSITY OF SINGAPORE

CS3211 - PARALLEL AND CONCURRENT PROGRAMMING

(Semester 2: AY2016/2017)

Time allowed: 2 hours

<u>**INSTRUCTIONS TO STUDENTS**</u> This assessment paper contains **FIVE (5)** sections totalling **FORTY (40)** marks, and comprises **TWELVE (12)** 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 (12)		
Speedup and analysis	Q2 (6)		
Accuracy and architecture	Q3 (6)		
Programming	Q4 (10)		
Models	Q5 (6)		
Total:	Q1-5 (40)		

Q1 (Short Answer Questions) (12 marks)

In the following short questions, each answer is worth 2 (TWO) marks.

1.1 Using Flynn's taxonomy (SISD, SIMD, MISD, MIMD), classify the following system and explain the reasoning for your choice: *The situation you are in right now, processing this test in parallel with other students.*

Answer:

1.2 Using Flynn's taxonomy (SISD, SIMD, MISD, MIMD), classify the following system and explain the reasoning for your choice: *The light-detector array in a digital camera.*

Answer:

1.3 Using Flynn's taxonomy (SISD, SIMD, MISD, MIMD), classify the following system and explain the reasoning for your choice: *Singapore's ASPIRE-1 supercomputer.*

Answer:

Q1 (Short Answer Questions) (Continued)

1.4 Using Flynn's taxonomy (SISD, SIMD, MISD, MIMD), classify the following system and explain the reasoning for your choice: *An IBM mainframe computer from the 1960s.*

Answer:

1.5 In a message-passing system, messages arrive at a processor at an average rate of 10^{10} bytes per second. The dynamic buffer that holds the arriving data averages 10^6 bytes in size. What is the average latency for processing the arriving messages, in seconds?

Answer:

1.6 Spectral methods are a particular kind of parallel programming architecture/paradigm. Briefly explain the most important communication characteristic of Spectral methods, giving an example.

Answer:

Q2 (Speedup and analysis) (6 marks)

A program that solves N linear equations with Gaussian elimination has been ported to run in parallel on P processors. The execution time is closely matched by the following formula:

$$T(N,P) = 10^{-9}N^2 + 10^{-7}\frac{N^2}{\sqrt{P}} + 10^{-11}\frac{N^3}{P} \text{ seconds}$$

2.1 Find the sequential execution time for a system of 10^5 equations. (1 mark)

Answer:

2.2 Find the Amdahl (fixed-size) speedup for a 4096-processor systems compared to a single processor. Show your working. (2 marks)

Answer:

Q2 (Speedup and analysis) (Continued)

2.3 By numerical methods (not algebra), find a value for N such that the execution time is close to that of the sequential execution time for 10^5 equations. (2 marks)

Answer:

2.4 How much time would a single processor need to run a problem of the size N found in (2.3)? (1 mark)

Answer:

Q3 (Accuracy and architecture) (6 marks)

3.1 An inexperienced programmer has a serial program that generates 100000 random real numbers between 0 and 1, and sums them. The programmer ports it to a parallel system with 64 processors, but the parallel system produces a different answer. Give three likely reasons for the difference. (3 marks)

Answer:

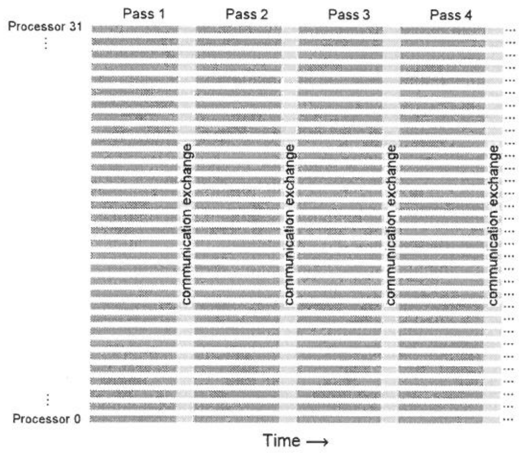
Q3 (Accuracy and architecture) (Continued)

3.2 One suggested way of improving MPI operations such as scan is to shift the aggregation operations into the switching/networking fabric. Briefly outline why such a technique would result in a speedup. (3 marks)

Answer:

Q4 (Programming) (10 marks)

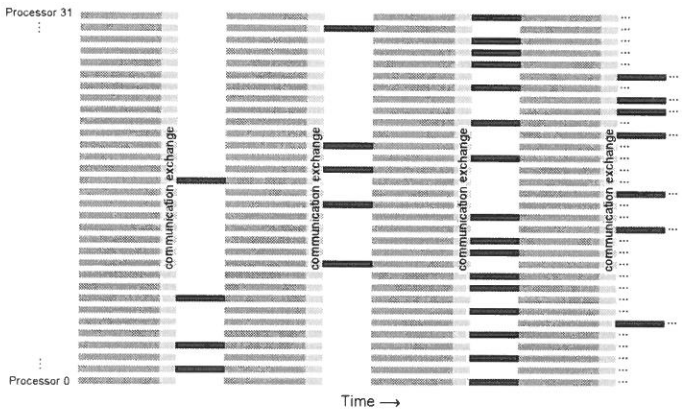
A parallel system with 32 processors performs a series of Fast Fourier Transforms (FFTs) where the computation alternates with an all-to-all communication exchange that must complete before the computation can proceed. The computation takes 5 seconds and the communication takes 1 second. Four passes of the series are shown below, but it continues with the same pattern:



4.1 Assuming a fixed problem size (Amdahl model) and that a serial processor would compute at the same speed but not have to do the communication exchanges, what is the predicted speedup of the 32 processors compared to a single processor? (2 marks)

Q4 (Programming) (Continued)

(Q4 continued) However, in practice the operating system must execute a daemon task to manage memory every 27 seconds. Each processor does this from an independent start time, resulting in a random pattern. The daemon takes each processor 3 seconds to finish, during which time it cannot do anything else. This stalls the entire communication exchange by 3 seconds, since at least one processor is always running the daemon task:



4.2 What is the actual speedup of 32 processors compared to a single processor, given the above pattern and assuming that a single processor must also run the daemon every 27 seconds? (2 marks)

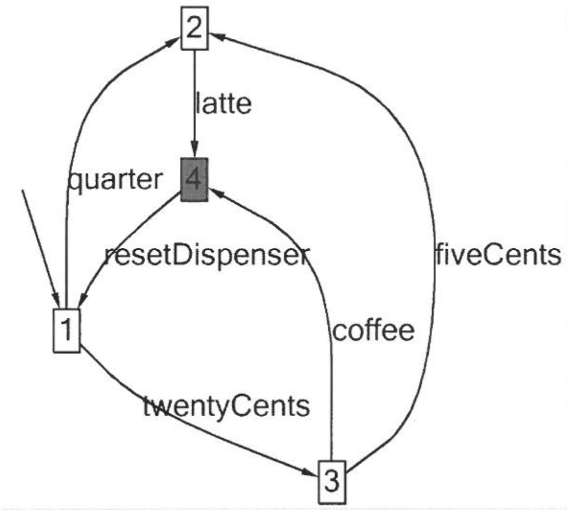
Q4 (Programming) (Continued)

4.3 Suggest a strategy to reduce the inefficiency caused by the system daemon. (3 marks)

4.4 What is the speedup, using the strategy you created in (4.3)? (3 marks)

Q5 (Models) (6 marks)

The following LTS describes the allowable events for a drinks dispenser, which dispenses a latte if given a quarter (25 cents), or coffee if given 20 cents.



5.1 Specify the drinks dispenser `Dispenser()` using a single process in CSP: (2 marks)

Note that you cannot use parallel composition. Your single process could be constructed from multiple interlocking process definitions for clarity however:

```
Dispenser() = .... -> B();
B()         = .... -> Dispenser();
```

Answer:

Q5 (Models) (Continued)

5.2 A student, modelled as the following CSP process, interacts with the drinks dispenser. Show/Draw the LTS for the parallel composition System() of Student() and Dispenser(). Draw your LTS in the box below. (2 marks)

```
Student() = hangout -> fiveCents -> twentyCents -> latte  -> Student();
System()  = Student() || Dispenser();
```

Answer:

5.3 Briefly explain how you could model, in CSP, MPI calls like MPI_Send and MPI_Isend. What differentiates them and how could you model them as processes? Give the CSP models. (2 marks)

Answer:

=== END OF PAPER ===