RHODES UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

EXAMINATION: NOVEMBER 2022

COMPUTER SCIENCE HONOURS
PAPER 1
DISTRIBUTED AND PARALLEL PROCESSING

Internal Examiner: Prof GC Wells
External Examiner: Dr D Vogts

MARKS: 120 marks
DURATION: 2 hours

GENERAL INSTRUCTIONS TO CANDIDATES

1. This paper consists of **3 pages** and **10 questions**. **Please ensure that you have a complete paper.**
2. Answer ALL questions.
3. The use of calculators is permitted in the examination, however, make sure that you show all workings.
4. The Oxford Concise English Dictionary **may** be used during this examination.

DISTRIBUTED AND PARALLEL PROCESSING [120 MARKS]

QUESTION 1: Theory [7 marks]

During the course the statement was made that "*this might be the most important course you study*". Discuss this claim, and the reasons for it.

QUESTION 2: Hardware/Theory [5+5 = 10 marks]

a) Discuss the relationship between *processor clock frequency* and *power consumption*, and the implications that this has had for modern processor design and the importance of parallel processing.

b) Discuss the relationship between the *computation : communication ratio* and *grain size* (or *granularity*) of a parallel application.

QUESTION 3: Hardware Topologies [12 marks]

During the course we discussed several different processor *topologies* (or network topologies, for example, the Simple 2-D Mesh network), and considered four criteria for assessing the benefits of different topologies: diameter, bisection width, number of edges, and edge length. Explain these four criteria, and state what the the ideal characteristics are for each criterion.

QUESTION 4: Theory [6 marks]

You have an application where 12% of the computation *cannot* be parallelised. What is the maximum speedup you might expect with 200 processors available, compared to a sequential version of the application using one processor (show your working)? State the source/name of the equation you use.

QUESTION 5: OpenMP [3+8+2+3 = 16 marks]

a) What is the effect of executing the following code fragment using OpenMP (i.e. what value is it calculating)?

```
int f = 1;
#pragma omp parallel for reduction(*:f)
for (int k = 1; k <= N; k++)
  { f *= k;
  }
```

b) Explain the bold line in the code above in detail.

c) How many threads will be created to execute the above code fragment? How is this number determined?

d) What are the advantages of using OpenMP compared to other forms of multithreading parallelism?

QUESTION 6 **[20 marks]**

EITHER: Multiprocessing

Describe how you would use the UNIX System V IPC *semaphore* and *shared memory* facilities in order to provide synchronous communication of simple messages between two processes. Your solution should allow a sender to place a message in a shared memory segment and then wait until the receiver has read it before proceeding. You need not give accurate code for your answer, but should mention the important steps in creating and using the IPC facilities that you need to use.

OR: Java Threads

Discuss the *executor* services that are provided by the Java Concurrency Utilities package (`java.util.concurrent`). What is the primary purpose of these services? You should include a discussion of the mechanisms provided to allow threads to return results.

QUESTION 7: CSP **[15 marks]**

Describe in detail how the Dining Philosophers' Problem might be solved using CSP. Include details of the processes and channels required, and a discussion of how deadlock is prevented in your solution.

QUESTION 8: occam **[10 marks]**

Explain what the following occam code does, and how it uses standard occam/CSP language features in an unusual way:

```
VAL INT Delay IS 200 :
INT Now :

Time ? Now
ALT
    Ch ? SomeThing
         -- do something useful
    Time ? AFTER Now + Delay
         SKIP
```

QUESTION 9: Remote Procedure/Method Calling **[12+2 = 14 marks]**

a) Explain in detail how *remote procedure/method calls* are implemented for distributed systems.

b) What is the role of the *registry* in Java's RMI?

QUESTION 10: CSP Metalanguage **[10 marks]**

A machine with alphabet {in50c, out20c, out10c} repeatedly gives change for 50c. The customer may choose any combination of sequences of 20c and 10c coins, provided that the total value equals 50c. Using the CSP metalanguage, construct the process CH to behave as described above.

END OF THE EXAMINATION