THE AUSTRALIAN NATIONAL UNIVERSITY
Mid Semester Examination – March 2015

COMP4300/8300
Parallel Systems

Study Period: 15 minutes
Time Allowed: 1 hour
Permitted Materials: Non-Programmable Calculator

The questions are followed by labelled, framed blank panels into which your answers MUST be written. Extra boxes are given at the end of the paper. Scribble paper is available, but no additional paper for marking will be provided. Writing outside of the boxes may not be marked.

This exam is marked out of 30. You should answer all questions.

Your mark for this exam will contribute 15% of your total course mark, according to the marking scheme given on the course web page.

Student Number:

The following are for use by your friendly examiner.

<table>
<thead>
<tr>
<th>Q1 Mark</th>
<th>Q2 Mark</th>
<th>Q3 Mark</th>
<th>Q4 Mark</th>
<th>Q5 Mark</th>
<th>Total Mark</th>
</tr>
</thead>
</table>
Question 1 [5 marks] General Motivation and Background

Moore’s law and Dennard scaling underpinned CPU development from the 1970s until around 2004.

(a) What is Moore’s law and what is Dennard scaling?

(b) Since 2004 Moore’s law has continued to be observed, but Dennard scaling is no longer observed. What is the implication of this, and how have processor designers responded to this limitation?

(c) In 1991 David Bailey wrote a paper entitled “Twelve ways to fool the masses when giving performance results on parallel computers”. List two ways in which you would attempt to trick me into believing that the parallel code you developed as part of your assignment is performing better than it really is.
Question 2 [7 marks] Parallel Hardware

A mesh of trees is a network that imposes a tree interconnection on a grid of processing nodes. A $\sqrt{p} \times \sqrt{p}$ mesh of trees is constructed by using one set of binary trees to connect all processors in each row and another set of binary trees to connect all processors in each column. This is illustrated below for a $4 \times 4$ mesh of trees:

(a) Give one advantage and one disadvantage of the mesh of trees network over a 2D torus.
(b) Outline (eg draw) how you would perform a broadcast operation on the mesh of tree architecture.

[2 marks]

(c) For a mesh of trees of size $\sqrt{p} \times \sqrt{p}$ and in terms of $p$ what is the i) bisection width; ii) diameter; and iii) total number of switching nodes.

[3 marks]
Question 3  [6 marks] Parallel Performance

You have access to a computer that has an 8 core processor and an attached accelerator (eg a GPU). You measure the elapsed time required to multiply one $1000 \times 1000$ matrix by another $1000 \times 1000$ matrix where both matrices contain double precision floating point numbers. You find the time required on ONE core of the 8 core processor is 24 seconds while running solely on the accelerator takes 1 second.

(a) In each of the two measured cases, what is the performance of the matrix multiplication in floating point operations per second?

\[
\text{Performance} = \frac{\text{Number of floating point operations}}{\text{Time}}
\]

[2 marks]

(b) You parallelise the matrix multiplication by assigning different rows of the first matrix to different cores on the 8 core processor. What elapsed time do you expect to observe when running your parallel code over 8 cores. Justify your answer.

\[
\text{Expected time} = \frac{\text{Total time}}{\text{Number of cores}}
\]

[1 mark]

(c) You extend the parallelization such that rows of the first matrix can be distributed to either one of the 8 cores of the main processor or to the accelerator. If using all 8 cores and the accelerator, how many of the 1000 rows would you allocate to each core and how many to the accelerator? (This should sum to 1000). Explain your decision. What do you expect the total execution time to be?

\[
\text{Rows per core} + \text{Rows to accelerator} = 1000
\]

[3 marks]
Question 4 [6 marks] Message Passing

Consider the following MPI code:

```c
#include <stdio.h>
#include "mpi.h"

int main(int argc, char* argv[]) {
    MPI_Status status;
    int rank, size, i, count=10, buf1[count], buf2[count];
    
    MPI_Init( &argc, &argv );
    MPI_Comm_rank( MPI_COMM_WORLD, &rank );
    MPI_Comm_size( MPI_COMM_WORLD, &size );
    
    for (i=0; i<count; i++){buf1[i]=0; buf2[i]=0;}
    
    if (rank == 0){
        for (i=0; i<count; i++){buf1[i]=i; buf2[i]=-1;}
        MPI_Bcast(buf1, count, MPI_INT, 0, MPI_COMM_WORLD);
        MPI_Send(buf2, count, MPI_INT, 1, 99, MPI_COMM_WORLD);
    } else if (rank == 1){
        MPI_Recv(buf2, count, MPI_INT, 0, 99, MPI_COMM_WORLD, &status);
        MPI_Bcast(buf1, count, MPI_INT, 0, MPI_COMM_WORLD);
    } else {
        MPI_Bcast(buf1, count, MPI_INT, 0, MPI_COMM_WORLD);
    }
    for (i=0; i<count; i++){printf("rank %d %d %d\n",rank, buf1[i], buf2[i]);
    
    MPI_Barrier(MPI_COMM_WORLD);
    MPI_Finalize();
    return 0;
}
```

(a) The `MPI_Recv()` function has seven arguments. Define each argument.
(b) In MPI there are collective and non-collective communication calls. Explain the difference. Which MPI functions in the above code are collective communications? 

[2 marks]

(c) Although as written this code runs to completion on the NCI Raijin system, it is NOT guaranteed to work with all MPI installations. Explain why this is the case, and how you might demonstrate this problem on Raijin.

[2 marks]
Question 5 [6 marks] Parallel Algorithms

(a) Consider the following vector operation

\[
\text{for (i=1; i<N; i++) } X[i] = X[i+offset] + X[i];
\]

Explain under what circumstances this code can run and give the correct answers on a SIMD computer.

For vector \( X[] \) a scan operation is defined as follows:

\[
\text{ScanX[0]=X[0];}
\]

\[
\text{for (i=1; i<N; i++) ScanX[i] = ScanX[i-1] + X[i];}
\]

(b) What is the difference between an inclusive and an exclusive scan operation? Which is given in the above code?
(c) For a vector of length 16 stored on a distributed memory parallel computer with one element per node, draw a diagram to illustrate how you would perform a scan operation capable of running in parallel.