THE AUSTRALIAN NATIONAL UNIVERSITY
First Semester Examination – June 2009

COMP4300/6430
Parallel Systems

Study Period: 15 minutes
Time Allowed: 3 hours
Permitted Materials: Non-Programmable Calculator

This exam is worth 40% of your total course mark. Exam questions total 160 marks, with marks awarded as integers according to the breakdown given.

Answer ALL questions.

Write your answers using a black or blue pen.

Clarity and conciseness in answers will be highly valued; marks may be lost for supplying irrelevant information.
**Question 1** [44 marks]

In this question we will use the following notation:

\[ W(var)\text{value} : \text{means write value to shared variable } var \]
\[ R(var)\text{value} : \text{means read shared variable } var, \text{obtaining value} \]

For processor \( N \), denoted as \( P_N \), we represent a series of memory read and write operations as

<table>
<thead>
<tr>
<th>Processor</th>
<th>First to last event list</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_N )</td>
<td>( R(x)0 ) ( W(x)1 ) ( W(y)2 ) ( R(z)2 )</td>
</tr>
</tbody>
</table>

where time proceeds from left to right. Thus, in the above processor \( N \) first reads shared variable \( x \) from memory and obtains a value of 0, it then writes 1 to shared variable \( x \), followed by writing 2 to shared variable \( y \), and then finally it reads shared variable \( z \) and obtains a value of 2.

(a) Define sequential memory consistency. Your answer should include one example of a series of events that conforms to sequential memory consistency, and one example of a series of events that does not.  

[10 marks]

(b) Modern computer hardware usually does not enforce sequential memory consistency. Explain why this is the case, and describe TWO different aspects of the hardware that can lead to non-sequential consistency.  

[10 marks]

(c) Define cache coherency and detail one cache coherency protocol.  

[12 marks]

(d) Explain why the following series of events are not sequentially consistent, but they could occur on a shared memory computer that uses a directory based cache coherency protocol. All variables are initialized to zero.

<table>
<thead>
<tr>
<th>Processor</th>
<th>First to last event list</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_0 )</td>
<td>( W(x)1 ) ( W(y)2 )</td>
</tr>
<tr>
<td>( P_1 )</td>
<td>( R(x)0 ) ( R(x)2 ) ( R(x)1 ) ( R(y)0 ) ( R(y)1 )</td>
</tr>
<tr>
<td>( P_2 )</td>
<td>( R(y)0 ) ( R(y)1 ) ( R(x)0 ) ( R(x)1 )</td>
</tr>
<tr>
<td>( P_3 )</td>
<td>( W(x)2 ) ( W(y)1 )</td>
</tr>
</tbody>
</table>

[12 marks]
Question 2 [16 marks]

The following ZPL program is taken from Chapter 8 of “Principles of Parallel Programming”, by Lin and Snyder. A copy of this chapter was available on the course web site.

```
1  program Life;
2  config const n : integer = 50;
3
4  region
5     R = [1..n, 1..n];
6     BigR = [0..n+1, 0..n+1];
7
8  var
9     TW:[BigR] boolean = 0; -- The World
10    NN:[R] integer; -- Number of Neighbors
11
12  direction
13    nw=[-1, -1]; no=[-1, 0]; ne=[-1, 1];
14    we=[ 0, -1]; ea=[ 0, 1];
15    sw=[ 1, -1]; so=[ 1, 0]; se=[ 1, 1];
16
17  procedure Life();
18  begin
19      --Initialize the world
20      [R] repeat
21          NN:=TW@nw+TW@no+TW@ne+
22          TW@we +TW@ea+
23          TW@sw+TW@so+TW@se;
24      TW:=(TW & NN = 2) | (NN = 3);
25      until !(<< TW);
26  end;
```

The [R] in line 20 defines a region. The @ operator is a translation, requiring an array and a direction as operands. The << symbol in line 25 is a reduction operator. The remaining syntax for ZPL is similar to C, or is self explanatory.

(a) Describe in words precisely what operations are being performed by i) lines 21-23, ii) line 24, iii) line 25. Where array data is being manipulated be sure to define the relevant array indices being manipulated.

[8 marks]

(b) What are the computational and communication costs associated with i) lines 21-23, ii) line 24, iii) line 25. Assume the code is executed on a distributed memory parallel computer with P processors.

[8 marks]
Question 3  [44 marks]

The following code is a pthread implementation of a bounded buffer used to store data of type Item.

```c
1 pthread_mutex_t lock=PTHREAD_MUTEX_INITIALIZER;
2 pthread_cond_t nonempty=PTHREAD_COND_INITIALIZER;
3 pthread_cond_t nonfull=PTHREAD_COND_INITIALIZER;
4 Item buffer[SIZE];
5 int put=0;
6 int get=0;
7
8 void insert(Item x)
9 {
10   pthread_mutex_lock(&lock);
11   while ((put+1)%SIZE == get)
12     {
13     pthread_cond_wait(&nonfull, &lock);
14   }
15   buffer[put]=x;
16   put=(put+1)%SIZE;
17   pthread_cond_signal(&nonfull);
18   pthread_mutex_unlock(&lock);
19 }
20
21 Item remove()
22 {
23   Item x;
24   pthread_mutex_lock(&lock);
25   while (put==get)
26     {
27     pthread_cond_wait(&nonempty, &lock);
28   }
29   x=buffer[get];
30   get=(get+1)%SIZE;
31   pthread_cond_signal(&nonfull);
32   pthread_mutex_unlock(&lock);
33   return x;
34 }
```
(a) Bounded buffers are often used in producer-consumer type problems. Outline one specific problem scenario where you would use a bounded buffer. (General or poorly defined scenarios will receive low marks.)

[6 marks]

(b) The above code uses both mutexes and condition variables to synchronize between threads. Explain the differences between these two pthread synchronization types, and give one other means by which threads are synchronized in pthreads.

[10 marks]

(c) Your colleague suggests that the while loops at lines 11 and 26 could be replaced by a simple if test. Why do you NOT agree with their suggestion?

[6 marks]

(d) When the buffer is full, how many data items does it hold? Explain how you derive your answer.

[8 marks]

(e) The code uses a single mutex (lock) to protect both the nonempty and nonfull condition variables. If this single mutex were replaced by two different mutexes what would be the advantages/disadvantages?

[8 marks]

(f) The pthread_cond_wait() routine takes the address of the protecting mutex as a parameter so that the routine can atomically block the waiting thread and release the lock that is held by the waiting thread. Explain in detail why these two operations must be performed atomically.

[6 marks]
Question 4  [16 marks]

(a) Describe the elements of MapReduce. Your reply should offer a short 1-paragraph overview of MapReduce, and then address each the following questions:

(i) How does one utilise MapReduce?
(ii) How is parallelism implemented?
(iii) How are data transfers accomplished?
(iv) What is a “Combiner” and why is it significant?

[10 marks]

(b) Given your knowledge of MapReduce, apply it to the following problem: Consider a set of total cloud cover data spread across multiple files, and within each file organised as a three-dimensional array with dimensions time, latitude, and longitude. The number of time samples in each file may vary, but for each time value cloud cover data is provided for the same set of latitude and longitude points. The cloud cover data values are integer percentages ranging from 0 to 100.

Describe how MapReduce could be employed to construct a global histogram from this data. Specifically create an array \( \text{counts}(0:100) \), where \( \text{counts}(i) \) is the number of instances of \( i\% \) cloud cover for all time measurements and all latitude and longitude data points. Describe what intermediate key and value pairs your \( \text{Map}() \) function emits, and how these intermediate \( \langle \text{key}, \text{value} \rangle \) pairs are processed by your \( \text{Reduce}() \) function to create the resulting histogram. Also detail how a \( \text{Combiner}() \) function might be implemented to optimise this process.

[6 marks]
Question 5 [40 marks]

Parallel computing systems are becoming increasingly heterogeneous. One heterogeneous system that was considered in this course was the IBM Cell Broadband Engine (CellBE). A diagram of CellBE system taken from the 2006 IEEE Micro article “Synergistic Processing in Cell’s Multicore Architecture”, by M. Gschwind, P. Hofstee, B. Flachs, M. Hopkins, Y. Watanabe, T. Yamazaki, (that was made available on the course web page) is given below.

In this diagram SMF is the synergistic memory flow controller, and SXU is the synergistic execution unit. The SXU is capable of executing 128 bit wide SIMD instructions. The local store on each SPU is 256kbytes, and is used to store both data and instructions. The clock rate of the CellBE is 3.2GHz.
In assignment 2 you implemented a parallel version of the Batchers sorting algorithm that sorted $N$ integers using $T$ threads on a multiprocessor shared memory system (machine \texttt{fremont}). In broad terms the algorithm involves the following steps

1. Divide the list of $N$ integers into $T$ sublists that are each assigned to a thread.
2. Each thread sorts its sublist using a conventional sorting algorithm such as quicksort.
3. Pairs of threads merge their sorted sublists. (After merging the sublist associated with each thread is ordered, with the largest element in the sublist on one thread being less than or equal to the smallest element in the sublist of the other thread.)
4. The above merging is repeated $\frac{1}{2} \lceil \log_2 T \rceil (\lceil \log_2 T \rceil + 1)$ times using different pairings of the threads in each repeat.

(a) Give a rough outline of the strategy you would use to implement Batchers sort on the CellBE system. You should use both the PPE and SPEs. Assume that you can use roughly half the available local store on each SPU to hold data (the rest is for instructions), and that the sorting problem size will exceed the aggregated local memory of all SPUs (i.e. you will need to transfer data back and forth between SPU local store and main memory on the PPE). If there are other parameters or details of the CellBE system that are important for your implementation, but that have not been detailed here, define what these are and make sensible assumptions accordingly.

(b) Discuss the performance you expect to obtain from your CellBE implementation when sorting a large number of integers (e.g. if you were to sort $2^{31}$ integers). Your answer should include as much detail as possible (given the information you have been provided with in this question and any assumptions you have made), and should comment on potential performance bottlenecks.