This question paper consists of 8 printed pages each of which is identified by the Code Number (COMP5811M)

School of Computing

January 2018

COMP5811M

Parallel and Concurrent Programming

Answer all 4 questions

Marks available: 60

Time allowed: 2 hours

Question 1.

(a) 5% of a program is inherently sequential; the remainder can be parallelised. Assuming you have 100 cores available, and ignoring communication costs, what speedup would be predicted (i) under Amdahl's law, and (ii) under the Gustafson-Barsis law? When you parallelise the program, how do you measure the observed speed-up?

[5 marks]

(b) Consider the piece of code given below, (i) identify the flow in the function `do_several_things()` (ii) explain why it does not work as the programmer expects in the comments, (iii) fix the flow.

```
void update_data(XData & d);
void process_xdata(XData & d);

void do_several_things(){
  XData data;
  //update data of type XData in a separate thread
  std::thread t (update_data, data);
  // do something else while data is being updated
  do_something_else();
  // to make sure data is fully updated
  // wait until the child thread joins
  t.join();
  // process updated data
  process_xdata(data);
}
```

[4 marks]

(c) Explain the Resource Acquisition Is Initialization (RAII) idiom and with one example explain how it can be used to simplify the challenge of concurrent programming.

[6 marks]

[question 1 total: 15 marks]

Question 2

(a) Suppose you have a block of code containing a critical section that can be accessed by only one thread at a time sandwiched between two other function calls that are thread-safe. The critical section is protected by two mutexes as follows:

```
std::mutex m;
std::mutex n;
preprocess();
std::lock(m, n);
critical();
m.unlock();
n.unlock();
postprocess();
```

Consider the following three scenarios. For each case, identify whether it is a programming error and describe the consequences.

i. Programmer A forgets to include the std::lock(m, n) call.
ii. Programmer B forgets to include the m.unlock() call.
iii. Programmer C swaps the m.unlock() and n.unlock() calls.

[6 marks]

(b) A programmer would like to implement a C++ class that models a thread safe stack. This thread safe class should allow for producer and consumer threads to communicate data. The producer threads push their new data on the stack, and the consumer threads pop the most recently added data from the stack. If a consumer thread attempts to pop from an empty stack, the thread should be suspended until a producer pushes data on to the stack. The partial code for this class is given below. Complete the implementation of `push()` and `wait_pop()` functions; include the necessary data members in your answer.

```
Template <typename T>
class tsstack{
 private:
   std::stack<T> data;
 public:
   void wait_pop(T & val);
   void push(T val);
}
```

Marks will not be deducted for syntactic mistakes, but it must be clear how the class would be implemented using the C++'11 multi-threading support.

[9 marks]

[question 2 total: 15 marks

Question 3

(a) The counter class given below is not thread safe. Can you make this class thread safe without using mutex? If your answer is yes, demonstrate how to do it, otherwise explain why not.

[2 marks]

```
class counter{
Private:
   int value;
Public:
   void increment(){
       ++value;
   }
  void decrement(){
      --value;
   }
};
```

(b) Explain the functionality of the compare_exchange_strong function from the C++ atomic library.

```
bool compare_exhange_strong(T & , T)
```

[5 marks]

(c) To implement a multi-threaded queue data structure, an existing structure (e.g. from STL) is wrapped by a C++ class, where a single mutex mediates access to member functions. Explain the drawback of this trivial solution on performance. Briefly explain two alternative approaches that have been used to avoid this problem in the implementation of multi-threaded data structures.

[3 marks]

(d) Assume you want to exploit a CUDA-enabled GPU to perform an add operation on two large vectors, returning the result in a third vector. The CUDA kernel function for the add operation is given below, and you need to write a C/C++ program to launch the kernel function on GPU. Describe the steps you need to take in your C/C++ program; if you need to call a library function, name the function. You do not need to write the actual code.

```
#define N ..
__global__
void add_kernel (int *a, int *b, int *c){
   int tid = blockIdx.x;
   If (tid < N)
    c[tid] = a [tid] + b [tid];
}
```

[5 marks]

[question 3 total: 15 marks]

Question 4

(a) A kernel is launched with a 1D grid containing 2D blocks of threads. Write the CUDA code that will calculate the unique thread id within each kernel instance.

[2 marks]

(b) Consider the following CUDA kernel:

```
//number of values to compute
#define N (2 << 20)
__global__
void foo(int n, int *res){
  int tid = blockIdx.x * blockDim.x + threadIdx.x;
  if (tid < N) {
    if (tid % 2)
      res[tid] = expensive_call_one(n, tid);
    else
      res[tid] = expensive_call_two(n, tid);
  }
}
```

When executed, this kernel will suffer a significant performance penalty. Identify what this penalty is, and explain why it occurs.

[3 marks]

(c) There are four types of memory available on a CUDA- enabled GPU. Briefly explain which one you would use in each of the following scenarios to get the best performance:

(i) Communication between threads within a block;

(ii) Data has to be visible to all threads (including the host), and last for duration of host allocation;

(iii) For the duration of the kernel, all threads in half-warps need to access the same element of a constant array data structure (value stored in the same address of memory).

[3 marks]

(d) A GPU has an architecture that supports 1024 threads/block, 2048 threads/SM, 16 blocks/SM and 32 threads/warp. Which of the following is the best configuration for thread blocks to execute a compute-intensive kernel: 8x8, 16x16, or 64x64? Justify your answer.

[4 marks]

(e) Name the current CUDA support for task parallelism and explain how it can be used to accelerate a computation. You may wish to use a diagram to support your answer.

[3 marks]

[question 4 total: 15 marks]

End