A30842 No calculator permitted in this examination

THE UNIVERSITY OF BIRMINGHAM

THIS PAGE TO BE REPLACED BY OFFICE

06 26945

Distributed and Parallel Computing

Summer 2017 $1^1/_2$ hours

[Answer ALL questions]

1. This question is about GPU programming and the CUDA programming model.

   (a) The NVidia GPU architecture and the CUDA programming model draw distinctions between different types of memories. For each of the following memory types, identify their relative speed of access (very slow to ultra fast), their scope (thread, block or grid) and their approximate size limits on typical GPUs (e.g. thousands, millions or billions of units or bytes per Thread, Block or Grid):

   (i) Registers
   (ii) Local memory
   (iii) Shared memory
   (iv) Global memory

   [10%]

   (b) Show how to derive the Gustafson-Barsis law (any equivalent mathematical derivation will suffice) and explain what it means in terms of scalability of programs through parallelisation. **[10%]**

   (c) Assume that a CUDA kernel function has been written that correctly executes a matrix-matrix addition using a 2-dimensional grid and 2-dimensional block configuration where each thread calculates the addition of one cell of the matrix result.

   (i) What would be the consequences if the size of the grid used were **SMALLER** than necessary for the matrices used? **[3%]**

   (ii) What would be the consequences if the size of the grid used were **LARGER** than necessary for the matrices used? **[3%]**

   (iii) If the matrices involved were N rows and M columns, and assuming that you want to use a 16×16 block configuration, write the lines of code that replace the "`// MISSING CODE`" comment in the CUDA C code below to calculate the appropriate grid configuration.

```
# define BLOCK_WIDTH 16
// MISSING CODE
dim3 dimBlock ( BLOCK_WIDTH , BLOCK_WIDTH ) ;
matrixAdd <<< dimGrid, dimBlock >>>(d_A, d_B, d_C, N, M) ;
```

   [4%]

(d) The following code shows the beginning of a one-dimensional CUDA C kernel, with a block size configured to be 1024 threads, that copies from Global memory into Shared memory. In particular, each thread is responsible for copying two floats from the Global memory array to the Shared memory array:

```
#define BLOCK_SIZE 1024
__global void kernel (float *X, float *Y , int len)
{
    __shared__ float XY [ BLOCK_SIZE * 2 ];
    int i = (blockIdx.x * blockDim.x + threadIdx.x) * 2 ;
    if (i < len)
        XY [2 * threadIdx.x] = X[i];
    if (i + 1 < len)
        XY [2 * threadIdx.x + 1] = X[i + 1];
    ...
}
```

This code works correctly but performs very poorly.

(i) Explain all sources of this poor performance and describe how to fix them. **[10%]**

(ii) Rewrite the code to fix the problems you identified. **[10%]**

2. This question is about distributed algorithms

   (a) In a system with three processes, p, q, and r, and five messages m_1 to m_5, the following events occur in the following real time order:

   t_1: p sends m_1 to q
   t_2: q sends m_2 to p
   t_3: q receives m_1 from p
   t_4: q sends m_3 to r
   t_5: r receives m_3 from q
   t_6: p receives m_2 from q
   t_7: r sends m_4 to p
   t_8: p sends m_5 to r
   t_9: p receives m_4 from r
   t_{10}: r receives m_5 from p

   (i) Draw a space-time diagram of this execution **[5%]**

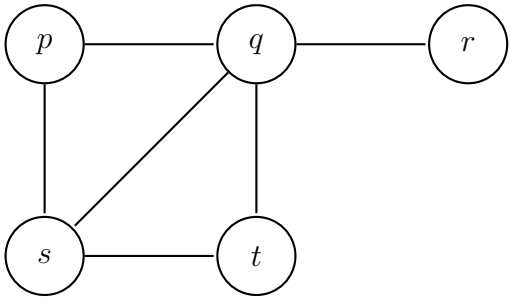
   (ii) Annotate each event on your space-time diagram by writing its Lamport clock value **below** each event **[7%]**

   (iii) Annotate each event on your space-time diagram by writing its vector clock value **above** each event **[8%]**

   (b) Explain, with the aid of a space-time diagram of an example execution, why the Chandy-Lamport global snapshot algorithm fails on a non-FIFO network **[5%]**

   (c) What two purposes does the control message in the Lai-Yang-Mattern global snapshot serve? **[5%]**

(d) Consider the following undirected FIFO network:



Taking p as the initiator, demonstrate a sample execution of the Echo traversal algorithm by annotating the channel lines as follows:

- Add message numbers to show the order in which the messages are sent. Any total ordering compatible with the algorithm is acceptable.
- Add arrows with the message numbers to show the direction in which the messages are sent.
- Indicate the resulting spanning tree by adding arrowheads to the corresponding channel lines to show the parent relation.

[10%]

(e) The Echo and the Tarry algorithms have a very similar structure, but they have different purposes and different details in their executions. Consider executing them simultaneously on identical large networks of processes.

(i) In general, which algorithm would you expect to finish first? Explain the reasons why. **[8%]**

(ii) Give an example of a network topology for which you would expect the execution times for the two algorithms to be roughly equal. **[2%]**