No calculator allowed in this examination

UNIVERSITY OF BIRMINGHAM

School of Computer Science

Distributed and Parallel Computing

Main Summer Examinations 2019

Time allowed: 1:30

[Answer all questions]

Turn Over

Note

Answer ALL questions. Each question will be marked out of 20. The paper will be marked out of 60, which will be rescaled to a mark out of 100.

Question 1

(a) (i) Explain what *Bank Conflicts* are in shared memory access in the CUDA architecture. **[2 marks]**

(ii) Explain how to predict the bank conflicts that a given section of CUDA kernel code will have. **[3 marks]**

(b) Consider a 2-dimensional CUDA kernel which is run with 32 times 32 threads per block and uses a shared memory matrix of size 32 times 32. For such a kernel, care is typically needed to avoid bank conflicts. A common strategy is to pad the matrix to size 32 times 33. Explain in detail why this strategy works to reduce bank conflicts without introducing new ones on typical matrix access patterns. **[5 marks]**

(c) Both the inclusive Blelloch and the inclusive Hillis-Steele-Horn algorithms calculate a *inclusive prefix sum* or *inclusive scan* operation. However, on arrays larger than the block size (typically 1024 words), they do not calculate a full scan, but only a block scan, where the cumulative sum is reset to zero at the start of each block.

Assume you need to execute a full inclusive scan in an efficient parallel manner, where the block size is 4 words and the segment size is 1 block (i.e. a block of 4 threads operates on a segment of 4 words only), on the following input array:

1	2	1	3	1	1	3	3	2	1	2	2	2	1	1	2

Display, in a single diagram:

- the other kernels you would need to write (you do not need to show the code or the parameters for these kernels),
- the order the kernels should be called in
- the actual numeric kernel configuration parameters for each kernel call and
- the result arrays of each kernel call for the given input

[10 marks]

Question 2

For this question, assume that primitive kernels are available for the normal and segmented versions of the operations of map, reduce, scan (inclusive and exclusive), compact, scatter and gather.

An efficient CUDA kernel is required to carry out large sparse matrix vector multiplications. The matrix, M, is stored in Compressed Sparse Row format, consisting of 3 vectors:

- V: non-zero values of M in top to bottom, left to right order,
- C: the column in M of the corresponding value of V and
- R: the index in V of each element that starts a row in M

Consider the following small example:

$$\begin{bmatrix} 0 & b & c \\ d & e & f \\ 0 & 0 & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} by + cz \\ dx + ey + fz \\ iz \end{bmatrix}$$

(a) Write out the values of V, C and R for the matrix of the small example above. **[5 marks]**

(b) Explain how the primitive kernels can be used to implement an efficient parallel version of the required large sparse matrix vector multiplication. **[10 marks]**

(c) In a diagram, show the values of all the vectors involved, including those of all the intermediate results on the small example above. **[5 marks]**

Question 3

(a) Lamport Clocks are *consistent* with causality whereas Vector Clocks are *strongly consistent* with causality. Explain what this difference means and, using a time-space diagram where the events are labelled with both Lamport and Vector Clock values, identify an example that demonstrates the difference. **[5 marks]**

(b) Draw a time-space diagram that shows an example of the execution of the Lai-Yang-Mattern global snapshot algorithm on two processes connected by a **NON-FIFO** bi-directional channel. Your example execution should include multiple base algorithm messages and message crossovers that illustrate the underlying NON-FIFO nature of the channel and how Lai-Yang-Mattern copes with it. All messages should be labelled: "W" for white, "R" for red and "C" for control, and any necessary parameters in the messages should be indicated. **[7 marks]**

(c) Explain two different ways to modify the Tarry traversal algorithm to ensure that it generates a depth first spanning tree and describe the differences that the two methods make to the space and time complexity of the algorithm. **[8 marks]**

This page intentionally left blank.

Distributed and Parallel Computing

Do not complete the attendance slip, fill in the front of the answer book or turn over the question paper until you are told to do so

Important Reminders

- Coats/outwear should be placed in the designated area.
- Unauthorised materials (e.g. notes or Tippex) must be placed in the designated area.
- Check that you do not have any unauthorised materials with you (e.g. in your pockets, pencil case).
- Mobile phones and smart watches **must** be switched off and placed in the designated area or under your desk. They must not be left on your person or in your pockets.
- You are not permitted to use a mobile phone as a clock. If you have difficulty seeing a clock, please alert an Invigilator.
- You are not permitted to have writing on your hand, arm or other body part.
- Check that you do not have writing on your hand, arm or other body part – if you do, you must inform an Invigilator immediately
- Alert an Invigilator immediately if you find any unauthorised item upon you during the examination.

Any students found with non-permitted items upon their person during the examination, or who fail to comply with Examination rules may be subject to Student Conduct procedures.