Parallel Programming, revision questions, 2019/20

Lecture 1, Introduction to Parallel Computing

- What is the goal of parallelisation?
- Explain the Moore's Law and its consequences on the development of parallel hardware.
- Explain the fact that while the transistor count in the processors is still rising, the clock rate trend has flattened since 2004/2005.
- Can we still improve the performance of our processors using frequency-scaling? Justify your answer.
- Provide two examples of applications which benefit from parallel computation. Could these problems be solved with standard serial hardware?

Lecture 2, Parallel Architectures

- Name the two independent dimensions used in Flynn's taxonomy to classify multi-processor computer architectures.
- Provide a brief definition of SIMD and explain what types of problems would benefit the most from SIMD implementations.
- Compare the pros and cons of shared and distributed memory architectures.
- Which category of Flynn's taxonomy do GPUs fall in?
- Is the GPU architecture optimised for low-latency applications or high-throughput applications, and why?
- What is heterogeneous computing?

Lecture 3, Patterns 1 - Map & Stencil

- What is a parallel pattern?
- What is a map pattern and its main characteristics? What is the theoretical and practical complexity of the map pattern?
- What is the strategy to perform parallel computation on large data inputs with a limited number of processing units?
- What is a stencil pattern and how can it be optimised?
- Which parallel patterns should be used for implementing the Scaled Vector Addition B = mA+B, where A,B are vectors and m is a scalar? What are the benefits of implementing this operation in a single kernel?

Lecture 4, Patterns 2 - Reduction

- Explain the reduce pattern and provide two example combiner functions for it.
- What are the requirements for the combiner functions in the reduce pattern?
- What is the span and work complexity of the reduce pattern?
- Why do we need thread synchronisation in the reduce pattern?
- Explain multi-pass reduction.
- Explain strategies for combining partial results from reduce operations performed by a number of work groups.

Lecture 5, Patterns 3 - Scan

- What is the difference between exclusive and inclusive scan?

- Explain the work efficient implementation of the parallel scan pattern.
- Explain the step-efficient implementation of the parallel scan pattern.
- What is the step and work complexity of both of these implementations? What problem size is most suitable for each of these implementations?
- Describe a general approach for scan which can deal with large data sets.
- Provide an example application of the scan pattern.

Lecture 6, Communication and Synchronisation

- Describe the memory hierarchy used in modern parallel hardware. What are the benefits of using local memory?
- How is private memory being used by individual threads/work items?
- Describe typical thread synchronisation mechanisms used in modern parallel hardware. What is the difference between local and global synchronisation? What are the memory barriers for?
- What are the atomic functions, their applications and limitations?

Lecture 7, Algorithms - Histogram

- What is a histogram and how its calculation can be parallelised using atomic functions?
- How does the input data distribution affect the performance of a parallel histogram implementation based on atomics?
- Describe a privatisation algorithm for calculating histograms. What is the difference between implementations using “per work item” and “per work group” local histograms? What are the limitations of both approaches?
- How does the sort-search algorithm for building histograms work? Which applications can benefit from using this approach?

Lecture 8, Algorithms - Sort

- What is a parallel equivalent of the bubble-sort procedure? How does it work? What is the step and work complexity of that algorithm?
- What is a bitonic sequence and how can it be constructed in parallel from an unordered input sequence?
- What is a bitonic split? Describe the sorting procedure for bitonic sequences.
- What is the complexity of bitonic sort?

Lecture 9, Algorithms - Search

- The binary search requires the input data to be sorted. What applications might justify that additional step?
- Describe the basic strategies for parallelisation of the search algorithms. How is the memory access pattern in parallelised linear and binary search affecting the performance?
- How does P-ary search exploit the parallelism?
- Compare the step complexity between P-ary and binary search.

Lecture 10, Data Reorganisation

- Describe how data reorganisation can help with parallelisation.
- What is the gather pattern?

- Given the following locations and source array, use a gather to determine what values should go into the output collection. Fill in the output array.

0	1	2	3	4	5	6	7	8	9	10	11	
3	7	0	1	4	0	0	4	5	3	1	0	Source array
1	9	6	9	3	Indices							
					Output array							

- What is the scatter pattern? Why does it require collision resolution?
- Describe how to implement histogram calculation using atomic functions.
- Fill in the output values in the following example, using the merge and priority scatter.

0	1	2	3	4	5	6	7	8	9	10	11	
3	7	0	1	4	0	0	4	5	3	1	0	input
2	4	1	5	5	0	4	2	1	2	1	4	locations
?	?	?	?	?	?	output						

- What is the benefit of exploiting coalesced memory access?

Lecture 11, Performance and Optimisation

- What is the difference between latency and throughput?
- What is the difference between speedup and efficiency in parallel programs?
- Which factors influence performance when parallelising sequential problems?
- Describe how to apply the Amdahl's Law to parallelisation of computer programs. What are the consequences of this law on the speedups arising from parallelisation?
- List the sources of overhead in parallel computation and summarise their influence on the performance.

Lecture 12, Libraries

- Compare low- and high-level parallel programming frameworks. Provide a suitable example of application scenario for each category.
- What are the main characteristics of modern heterogeneous computing parallel frameworks such as OpenCL?
- What are the benefits of modern parallel software libraries? Can you think of any limitations related to their use?