Test 1: Memo

EEE4084F

2016-03-08

Question 1: Data Partitioning **[13 Total]**

#1 The CPU has 4 cores, so I would use 4 threads (2 marks). More threads result in greater task-switching overhead (1 mark) – the other threads are all idle, so using more threads will not be able to obtain any more CPU-time from the scheduler. Fewer threads result in cores that are idle (2 marks). **[5]**

#2 I would partition the data such that each core is given 256 contiguous rows, but there are other options as well. In essence, you have to make optimal use of the cache, which is much faster than RAM. Both the input and output must fit in the cache of each core, in order to minimise cache misses. A partition size smaller than 64 kiB (i.e. 64 rows) would be sub-optimal. Anything larger would be good, up to a maximum of 256 rows. A partition size larger than 256 rows would result in some threads doing more work than others.

Any option where the student argues for column-partitioning is sub-optimal, as the data is row-ordered. Since the image size is known, static partitioning will be faster than dynamic partitioning.

The student only needs to argue for one case. The figure counts 2 marks, a sensible partitioning method 2 marks, and the explanation / argument the other 3 marks. **[8]**

Question 2: Landscape of Parallel Computing **[14 Total]**

#1 The paper lists a whole bunch of old vs. new wisdoms. Listing any 4 of these would be sufficient (2 marks each: one for the conventional wisdom and one for the new wisdom).

The other 2 marks are for explaining the trend. **[10]**

#2 A dwarf describes a set of applications (1 mark) that share common requirements (1 mark) and design patterns (1 mark). It was necessary to add additional dwarfs because the new applications emerged that did not fit in the original 7 (1 mark). **[4]**

Question 3: Parallel Computing and Benchmarking **[8 Total]**

#1 Any two points of the following: **[2]**

- The parallel version could potentially be faster than the original sequential solution.
- Often, a parallel solution (especially on FPGA) uses less power for the same speed performance.
- It's sometimes used for redundancy and fault tolerance.
- In some cases, a lot of little cores can provide the same speed performance while using fewer resources (transistors).
- Parallelism can be used to implement a decentralised processing solution.

#2 A golden measure is a version of the program that is known to produce correct results (1 mark). It is often sub-optimal, but uses easy-to-implement algorithms (1 mark).

The speed-up factor is defined as (2 marks): **[4]**

$$\text{speedup} = \frac{\text{runtime of golden measure}}{\text{runtime of the implementation under test}}$$

#3 The Von Neumann architecture has only one memory interface, so the program and data reside in the same memory-space (1 mark). The Harvard architecture uses separate memory for data and instructions (1 mark). **[2]**