Test 1: Lectures 2 to 8

EEE4084F 2018-03-15

Instructions:

- Answer on separate pages.
- Make sure that your student number is on all your answer pages.
- There are **4** questions, each divided into sub-questions. Answer all questions.
- Total time: 45 minutes.
- Total marks: 54.
- *PS: there an amazing benefit of a pthreads cheatsheet on the last page, woohoo! (There also another surprise on the last page in case you didn't study).*

Question 1: Landscape of Parallel Computing [20 Total]

These question relates to Seminar 1.

(a) The Berkeley paper uses the Golden Gate (illustrated right) to suggest a tension between two computer design approaches (which, when we look at the concept of HPEC, are essentially a merger of these) – what are these two approaches? [2]

(b) The Berkeley's Landscape of Parallel Computing discusses how processor architectures could follow a design of using thousands of processor cores per chip. But is this a generally a recommendable approach or is it more likely a special case. Argue for or against this approach and mention situations where this approach may be better and where it may be worse. (Note there is not necessarily a clear-cut answer for this, it is more your logical explanation that will be marked. You are welcome to use rough sketches if you think it needed.) [7 marks]

(c) You are no doubt well prepared to talk about DWARFs and possibly tell tales of little men inside your computer too. But in the Berkeley's Landscape paper, a DWARF is used in a more technical sense – define a DWARF according to how it is used in the Berkeley paper. Why is it that getting DWARFs (in the Berkeley sense to be clear) to do things could be a quicker and less error-prone than doing things from scratch? What are the benefits of using the DWARF approach? And why may it be more useful to benchmark a platform according to DWARF performance rather than using a long-serving and well-understood benchmarking library like Linpack? [7 marks]

(d) What is the difference between old conventional wisdom and new conventional wisdom? Give an example of one old convention and a corresponding new convention in terms of parallel computer system design. [4 marks]

Question 2: Speeding up processing using parallelism **[24 Total]**

Answer all following questions…

(a) Imaging that you did a GPU implementation for which you found it took 158ms to run and that the speedup over the serial golden measure (GM) was 10.56. You've forgotten what speed the GM ran at and now you've just shut down your PC – but that's clearly no problem; quickly calculate what the sequential speed was (show your working!). [4 marks]

(b) Consider that you have an array **x** in C that is of length **n** floats. You have already written a golden measure called ***midpoint*** in OCTAVE as shown below, which you are planning to convert into a thread-based function in C.

```
function y = midpoint (x)
  % calculate the midpoint of x
  y = (max(x) - min(x))/2;
 endfunction
```

i) First off, to check that you remember some OCTAVE, what would the following command sequence provide assuming that the above midpoint code is written in a file midpoint.m in the current directory?

```
x = [1:2:6] .* 8;
midpoint(x)
```

[4 marks]

ii) Explain how you would convert the ***midpoint*** function above into a pthread application. You need to explain how you would get the vector **x** and length **n** to the thread function, how you would partition the data and how you would implement the thread so as to divide the processing efficiently. (You can just write C code or you can explain in English with diagrams if needed to illustrate).

[12 marks]

iii) If you ran the following on OCTAVE:

```
tic; y=midpoint(rand(1,1000000)); toc
Elapsed time is 0.020000 seconds.
```

Which was using a single thread on an i7. How much faster do you think it might be if you run ran your solution to (ii) using 4 threads on the i7? Provide some calculation and motivation.

[4 marks]

Question 3: Parallel structures **[10 Total]**

(a) We discussed base core equivalents (BCE) back in lecture 4. Explain briefly what a BCE is and why it can be considered as taking Amdahl's further making comparisons fairer. To help the explanation you can make up two BCE examples (e.g. a 1x1 BCE and a 4x4 BCE).

[6 marks]

(b) What is meant by the term granularity? If a process had to do a lot of communication for calculating one result, would this be called fine or course grained – explain your answer. (PS: granularity, or *gran-u-halarity* does not mean your gran just cracked a good joke ☺)

[4 marks]

Bonus: what was the lawnmower scenario aimed to show? if one is assuming it wasn't just about moaning about how difficult it is to explain things to one's neighbours.

END OF TEST

CHEET SHEET

```
int pthread_create(pthread_t *tid, const pthread_attr_t *attr, void
*(*start_routine)(void*), void *arg);   // attr can be set to NULL for default attributes

void pthread_join(pthread_t thread_id, void **value_ptr);

void pthread_exit(void *value);

int pthread_kill(pthread_t thread_id, int sig);

int pthread_mutex_init(pthread_mutex_t *mutex, const pthread_mutexattr_t *attr);

int pthread_mutex_destroy(pthread_mutex_t *mutex);
```

For those who are bored or didn't study, here's a little finger labyrinth, should get to the centre quite safely.