CSCI-GA.3033-017 Special Topic: Multicore Programming

Homework 1

Due October 2, 2017

Please solve the following and upload your solutions to your private GitHub repository for the class as homework1.pdf by 11:59pm on the due date above. If for some reason this poses a technical problem, or you wish to include diagrams that you don't wish to spend time drawing in a drawing application, you may hand in a printed copy (*not* hand-written) at the beginning of class (6:20pm) on the day of the deadline. **Unlike labs, late homeworks will be assigned a grade of 0.**

---

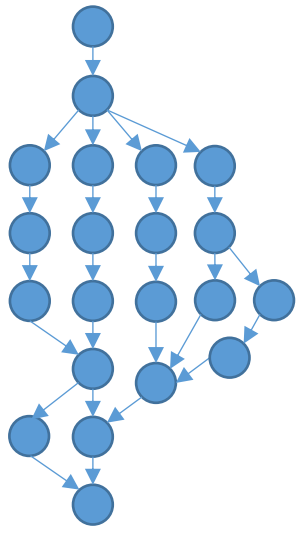
1. Warmup: Discuss five types of parallelism (hint: the lowest-level type is instruction-level parallelism). What are they, and for each one, what is necessary from the hardware and/or programmer? There are more than five types, but make sure you get most or all of the types that involve a single computer.

2. Without Googling (although the slides are fair game), what makes MISD processor architectures unusual? What are they generally used for, and why aren't they used more widely?

3. Apply Amdahl's law to compute the speedup for the following program if you have (a) 1, (b) 2, (c) 4, (d) 8, (e) 16, and (f) ∞ CPUs. In the following diagram, **S** portions are sequential and **P** are parallelizable.

S	P	S	P	P	S
15%	20%	10%	15%	30%	10%

4. Assuming that each block of the diagrammed program takes 1 unit of time, what is the work and span of the following program? What is the parallelized execution time T_P on (a) P=1 processors? (b) P=5 processors? (c) P=6 processors? (d) How long is the critical path?



5. Explain the difference between concurrency and parallelism with an example: if an operating system is executing two long-running programs, how would its scheduler execute the programs concurrently on one core, concurrently on two cores, or in parallel on two cores? Comment on running the programs in parallel on one core.

6. Broadly, what's the point of cache? Does its purpose differ between single- and multi-core processors (if so, how)? Does its implementation differ between single- and multi-core processors (if so, how and why)?

7. What are temporal and spatial locality, for example as they apply to caches? Which one(s) is/are more likely to come into play with frequent reads and writes to the elements in a 1x1-element matrix? A 1000x30-element matrix?