Final Exam: Questions

(10 points) What is GPU programming? What are the strengths of GPU programming over CPU programing?

GPU programming is about massive parallelism. Any program that requires thousands (or more) iterations of identical tasks is perfect for GPU programming - for example, computing matrices. On the other hand, GPUs are not as efficient with complex calculations especially regarding branching and looping. Furthermore, the memory model greatly limits GPU programming; any data that is processed, must be copied from the CPU memory into the GPU memory, loaded, executed, and then the results are copied back into the CPU memory.

(10 points) An oceanographer gives you a serial program and asks you how much faster it might run on 8 processors. You can only find one function amenable to a parallel solution. Benchmarking on a single processor reveals 80% of the execution time is spent inside this function. What is the best speedup a parallel version is likely to achieve on 8 processors using Amdahl's law?

If we consider the formula for maximum speed up is $max = 1/((1 - P) + P/S)$ where P is the part amendable to parallelization (thus $(1 - P)$ is the part not amendable to parallelization) and S is improvement factor. Plugging in the values and solving gets us:

$$max = \frac{1}{(1 - P) + \frac{P}{S}} \quad \textit{Amdahl's law}$$

$$max = \frac{1}{(1 - 0.80) + \frac{0.80}{8}} \quad \textit{Substitution}$$

$$max = \frac{1}{0.20 + 0.10} \quad \textit{Simplification}$$

$$max = \frac{1}{0.30} \quad \textit{Addition}$$

$$max = 3.3\overline{3} \quad \textit{Simplification}$$

Therefore, we could expect a maximum speed up of $\approx 3.3\overline{3}$x from executing a parallel version of this program on 8 processors using Amdahl’s Law.

For a problem size of interest, 6 percent of the operations of a parallel program are inside I/O functions that are executed on a single processor. What is the minimum number of processors needed in order for the parallel program to exhibit a speedup of 10?

If we consider the formula for maximum speed up is $max = 1/((1-P)+P/S)$ where P is the part amendable to parallelization (thus $(1-P)$ is the part not amendable to parallelization) and S is improvement factor. Plugging in the values and solving gets us:

$$max = \frac{1}{(1-P)+\frac{P}{S}}$$ *Amdahl's law*

$$10 = \frac{1}{(1-0.94)+\frac{0.94}{n}}$$ *Substitution*

$$10 = \frac{1}{0.06+\frac{0.94}{n}}$$ *Subtraction*

$$\left(0.06 + \frac{0.94}{n}\right) \times 10 = \left(\frac{1}{0.06+\frac{0.94}{n}}\right) \times \left(0.06 + \frac{0.94}{n}\right)$$ *Multiplication*

$$10\left(0.06+\frac{0.94}{n}\right) = 1$$

$$0.6 + \frac{9.4}{n} = 1$$ *Multiplication (by 10)*

$$\frac{9.4}{n} + 0.6 = 1$$
$$-0.6 \quad -0.6$$ *Subtraction*

$$\frac{9.4}{n} = 0.4$$

$$n \times \left(\frac{9.4}{n}\right) = (0.4) \times n$$ *Multiplication (by n)*

$$9.4 = 0.4n$$

$$\left(\frac{9.4}{0.4}\right) = \left(\frac{0.4n}{0.4}\right)$$ *Division (by 0.4)*

$$23.5 = n$$

Since processors do not come in halves, the minimum number of processors needed for the parallel program to exhibit a speedup of 10 is 24 processors.

(20 points) Average memory access time can be calculated using AMAT = Hit time + Miss rate x Miss penalty.

(10 points) If a direct mapped cache has a hit rate of 95%, a hit time of 4 ns, and a miss penalty of 100 ns, what is the AMAT?

$$AMAT = H_t + M_r \times M_p$$

$$AMAT = 4 + (1 - 0.95) \times 100$$

$$AMAT = 4 + (0.05) \times 100$$

$$AMAT = 9$$

(10 points) If replacing the cache with a 2-way set associative increases the hit rate to 97%, but increases the hit time to 5 ns, what is the new AMAT?

$$AMAT = H_t + M_r \times M_p$$

$$AMAT = 5 + (1 - 0.97) \times 100$$

$$AMAT = 5 + 0.03 \times 100$$

$$AMAT = 8$$

(50 points) Programming Assignment:

See attached folder for Programs, benchmarks are below.

PROBLEM	**NP/T = 1**	**NP/T = 2**	**NP/T = 3**	**NP/T = 4**
SEQUENTIAL	2.6758	-	-	-
MPI	2.6398	1.5704	1.1091	0.8845
OMP	1.7584	1.6584	1.5864	1.6523