CS 5220 – 2015-09-08 Preclass Questions

Michael Whittaker (mjw297)

September 7, 2015

1. For all i, j, computing $C[i, j]$ requires reading $2N$ double precision numbers each of which is 8 bytes. This is a total of $16N$ bytes. If $16N$ is much bigger than the size of the cache, then every time we read in a value, we must bring it into cache from memory. We perform a single flop for every pair of numbers, so this leads to a flop per 16 bytes, or an AI of $\frac{1}{16}$.

2. The matrix multiplication algorithm iterates over rows of A and then iterates over columns of B. If our cache is larger than $16N$, we can cache the rows of A. Since the cache is smaller than $8N^2$, we won't be able to cache the columns in B. We still perform a flop per pair of values, but now one of these values is cached. This leads to an AI of $\frac{1}{8}$.

3. Now we can read in all $16N^2$ bytes of A and B and perform N^3 operations. Then, we write out C which requires an additional $16N^2$ memory writes for an AI of $\frac{N^3}{32N^2} = \frac{N}{32}$.

4. I'll assume we're finding the biggest N such that the caches can fit all of A, B, and C. The largest N for 32 KB, 256 KB, and 6 MB are 105, 296, and 1449 respectively. This leads to arithmetic intensity of 3.28, 9.25, and 35.9 respectively.

5. Your CPU can perform $4 \times 8 \times 2.4 = 78.6$ Gflops/s. This means that an arithmetic intensity of $\frac{78.6}{25.6} = 3$ will make the computation CPU bound.

6. Matrix multiplication will be CPU bound for $N > 96 = 3 * 32$.

7. Flops/second will increase with N until N becomes 96, at which point, the CPU becomes saturated and Flops/second will plateau.