CS 5220 – Sep. 08 Preclass Questions

Eric Gao – emg222

Question 1

The memory-based arithmetic intensity is defined as:

$$\text{AI} = \frac{\text{\# flops}}{\text{\# bytes transferred between memory and cache}}$$

In the innermost loop, we have to perform 2 floating point operations on every iteration, which results in $2N$ flops for the inner loop. The total memory needed to perform the dot product between row i of A and column j of B is $2N \times 8$ bytes $= 16N$ bytes. Because this significantly larger than the size of our L3 cache, we can safely assume that under LRU, we need fetch all $16N$ bytes of data, everytime the innermost loop is executed. Therefore our memory-based AI is:

$$\text{AI} = \frac{1}{8}$$

Question 2

As shown in the previous question, we need $16N$ bytes of data to perform the innermost loop. Therefore, we look at the next most inner loop. Top perform the first inner loop (for (j = 0; ...)), we need $8N^2 + 8N$ bytes of memory. We need every entry from B which equals $8N^2$ bytes. We also need $8N$ bytes to keep row i of A around. To perform the first inner loop, we do $2N^2$ flops. The arithmetic intensity is now approximately:

$$\text{AI} = \frac{1}{4}$$

Question 3

We need to keep all of A, B, and C in cache, which equals $24N^2$ bytes. We then need to write to all of C back to main memory, which results in another $8N^2$ memory operations. This gives a total of $32N^2$ bytes transferred between memory and cache. We then have a total of $2N^3$ flops. Therefore, our arithmetic intensity is:

$$\text{AI} = \frac{N}{16}$$

Question 4

The L1 cache has 2^{15} bytes. We can just solve for N:

$$\begin{aligned} 24N^2 &\leq 2^{15} \\ N &\leq \sqrt{1365.33} = 36.95 \end{aligned}$$

N must be a positive integer, so we choose N to be 36. Because all of A, B and C can fit into the L3 cache, we can just use the formula obtained in Question 3 to obtain the AI.

$$\text{AI}_{L1} = \frac{1(36)}{16} = 2.25$$

The L2 cache has 2^{18} bytes. Solving again for N:

$$\begin{aligned} 24N^2 &\leq 2^{18} \\ N &\leq \sqrt{10922.66} = 104.5 \end{aligned}$$

We therefore choose N to be 104. The AI is then given by:

$$\text{AI}_{L2} = \frac{1(104)}{16} = 6.5$$

The L3 cache has on the order of 6 million bytes:

$$\begin{aligned} 24N^2 &\leq 6000000 \\ N &\leq \sqrt{250000} = 500 \end{aligned}$$

We therefore choose N to be 500. The AI is therefore:

$$\text{AI}_{L3} = \frac{1(500)}{16} = 31.25$$

Question 5

We first calculate the FLOPs/s on the CPU:

$$2\frac{\text{flops}}{\text{FMA}} \times 8\frac{\text{FMA}}{\text{cycle}} \times (2.4 \times 10^9)\frac{\text{cycles}}{\text{second}} \times 4 \text{ cores} = 153.6 \text{ GFLOPs / s}$$

To calculate arithmetic intensity, we divide FLOP/s by the memory bandwidth:

$$\text{AI} = \frac{153.6 \text{ GFLOPs / s}}{25.6 \text{ GB / s}} = 6 \text{ FLOPs / byte}$$

Question 6

We can solve for N:

$$6 = \frac{N}{16}$$

$$96 = N$$

When $96 \leq N \leq 500$, the naive matmul is CPU-bound.

Question 7

When $96 \leq N \leq 500$, the naive matmul is CPU-bound. Otherwise, the operation is memory bound. So when $N < 96$, the plot of Flops/s will be linearly increasing. The slope of this linear increase is:

$$slope = \frac{(25.6 \times 10^9)}{16} = 1.6 \times 10^9$$

This will be linearly increasing until we hit 153.6 Flops / s at $N = 96$. This line will then be straight until $N = 500$. The naive matmul will become memory-bound at this point. The Flop/s will decrease slowly at first, faster when the cache is smaller than $8N^2$, and faster still when the cache is smaller than $16N$.