CS 5220 Applications of Parallel Programming	Fall 2015
Kenneth Lim (kl545)	Intro. to Parallel Machines and Models

1. The number of double precision numbers evaluated in the innermost loop is $N+N = 2N$. Each double precision number is 8 bytes, thus the total number of bytes handled in the innermost loop is 16N. Since this value is larger than the size of the L3 cache, it must be paged in from memory. Thus the memory-based AI of the code is 1/8.

2. Since the cache is smaller than $8N^2$, it is infeasible to hold the entire matrix multiplication (exactly $8N^2$ bytes) in cache. However, we can easily hold one set of values in cache, say, a row of A for which all columns of B calculated. This improves the arithmetic intensity by a factor of 2, i.e. 1/4.

3. We start by paging in $16N^2$ bytes accounting for the matrices A and B. The number of operations performed is $2N^3$. This produces C, which has to be transferred back to memory, i.e. another $16N^2$ bytes. Therefore the AI is:

$$\frac{2N^3}{16N^2 + 16N^2} = \frac{N}{16}$$

4. The assumption used here is that C is not held in the cache because the result can be pushed back to memory as it is computed. Thus we need to store $16N^2$ bytes in cache. For the L1 cache, this corresponds to $N = 32$ with an AI of 2. For the L2 cache, this corresponds to $N = 90$ with an AI of 5.625. For the L3 cache, this corresponds to $N = 443$ with an AI of 27.6875.

5. The Flop rate for the machine is $4 * 2 * 8 * 2.4 = 153.6\,\mathrm{GFlops/s}$ (where each FMA instruction is 2 flops). Thus the max AI is $153/25.6 = 5.98$.

6. When $N = 16 * 6 = 96$.

7. Flops increases monotonically with N and asymptotes when the CPU-bound AI is reached.