Aditya Dhulipala
EE/CSCI 451, Spring 2015
Homework 1 Submission

1.

1. **Superscalar processor** – A processor which has the ability to execute multiple instructions in the same clock cycle.
2. **Row major layout** – Data layout ordering wherein contiguous blocks of memory are loaded with contiguous rows of the data (matrix) (as opposed to column major layout where contiguous columns are stored)
3. **Cache pollution** – The situation wherein data fetched into the cache is unnecessary for the execution of the program, i.e. cache could've been used to load useful data to hide memory latency instead.
4. **Instruction level parallelism** – Parallelism obtained by executing sequence of low-level instructions simultaneously through the use of hardware & software techniques such as dynamic instruction issue (hardware) or compilers(software).
5. **Cache line** – Unit of data fetched from main memory to cache
6. **Data dependency** – The issue where the result of one particular instruction may be required to execute some subsequent instruction.
7. **Very Long Instruction Word (VLIW) Processor** – Processors that can execute a group of instruction packed into a single long instruction word. (Used to exploit instruction level parallelism by resolving dependencies through software techniques).
8. **Spatial locality** – A computation-centric view of memory access wherein the consecutive words in memory are used by successive instructions
9. **Single instruction multiple data** – A parallel computing architecture where one instruction is executed by all the processing elements on all the data (each processing element works on one piece of the data). This execution is performed in lockstep operation.
10. **Implicit parallelism** – Parallelism inherent in the program because of the nature of operations/computations being performed.

2.

1. For the given symmetric multiprocessing system, effective memory access time is

$$0.8 * 10 + 0.1 * 100 + 0.1 * 400 = 58ns$$

This means the systems takes $58ns$ to fetch 1 word from memory (including cache). Assuming we perform 1 $operation/word$, the peak computation rate is

$$\frac{1}{58} * 10^9 = 17.24MFLOPS$$

2. For the given cache hit ratio in a single processor system, effective memory access time is

$$0.7 * 10 + 0.3 * 100 = 37ns$$

Assuming the system performs 1 $operation/word$ as before, the peak computation rate is

$$\frac{1}{37} * 10^9 = 27.02MFLOPS$$

$\therefore$ The fractional computation is rate is 17.24/27.02

3. 1. Latency of DRAM is 100 cycles, i.e.

$$= 100 * \textit{clock cycle of processor} = 100 * 1ns$$

To execute the instruction in the for-loop once we need to fetch 2 words. One word of matrix A and one of B.

$$\begin{aligned} \textit{Fetch one word of } A &\rightarrow 100ns \\ \textit{Fetch one word of } B &\rightarrow 100ns \\ \textit{Execute 1 multiply-add operation, i.e. 2 FLOPs} &\rightarrow 1ns \\ \textit{2 FLOPs in} 201ns & \end{aligned}$$

$$\begin{aligned} \implies \textit{peak achievable performance} &= \frac{2}{201} * 10^9 FLOPS \\ &= 9.95 MFLOPS \\ &\approx 10 MFLOPS \end{aligned}$$

2. The processor fetches four words in each memory cycle. So, the processor fetches 2 words of matrix A and 2 words of B in one cycle.

$$\begin{aligned} \textit{Fetch 2 words of A and 2 words of } B &\rightarrow 100ns \\ \textit{Execute 2 multiply-add operation, i.e. 4 FLOPs} &\rightarrow 1ns \\ \textit{4 FLOPs in} 101ns & \end{aligned}$$

$$\begin{aligned} \implies \textit{peak achievable performance} &= \frac{4}{101} * 10^9 FLOPS \\ &= 39.60 MFLOPS \\ &\approx 40 MFLOPS \end{aligned}$$

4. The latency of DRAM is $100 * 0.5ns = 50ns$. Assume that the cache is of size $16KB$. This is enough to fit the vector. The processor first reads the vector from DRAM into the cache.
This takes $4K * 50ns = 200\mu s$
We don't need to consider this into the computation rate since this is a one-time operation. Henceforth, the vector data will be loaded directly from the cache.

$$\begin{aligned} \textit{Fetch 2 words of matrix} &\rightarrow 50ns \\ \textit{Fetch 2 words of vector} &\rightarrow 0.5ns \end{aligned}$$

The processor issues both the above instruction at the same time since it can issue 4 instructions in each cycle. This means that the DRAM latency hides the cache latency,
i.e. effective latency to load operands is 50ns.

$$\begin{aligned} \textit{Execute 1 multiply-add operation, i.e. 2 FLOPs} &\rightarrow 0.5ns \\ \textit{2 FLOPs in } 50.5ns & \end{aligned}$$

$$\begin{aligned} \implies \textit{peak achievable performance} &= \frac{2}{50.5} * 10^9 FLOPS \\ &= 39.60 MFLOPS \\ &\approx 40 MFLOPS \end{aligned}$$