NYU Tandon School of Engineering

Fall 2024, ECE 6913

Homework Assignment 1

Instructor: Azeez Bhavnagarwala, email: ajb20@nyu.edu

Course Assistant Office Hour Schedule

On Zoom: 9:30AM – 11AM Monday, Tuesday, Wednesday, Thursday, Friday

Join Zoom: https://nyu.zoom.us/j/99424200816

1. Aishwarya Kandam, ak11049@nyu.edu
2. Aditya Krishna, ak11137@nyu.edu
3. Srinatha Sivareddy, ss18364@nyu.edu
4. Manoj Srinivasan, ms14845@nyu.edu
5. Aditya Ojha, ao2612@nyu.edu

Homework Assignment 1 [released Sunday Sept 8th 2024] [due Friday Sept 20th 11:59PM]

You *are allowed* to discuss HW assignments with anyone. You are *not allowed* to share your solutions with other colleagues in the class. Please feel free to reach out to the Course Assistants or the Instructor during office hours or by appointment if you need any help with the HW.

Please enter your responses in this Word document after you download it from NYU Classes. *Please use the Brightspace portal to upload your completed HW.*

1. Assume a program requires the execution of 60×10^6 FP instructions, 120×10^6 INT instructions, 60×10^6 L/S instructions, and 16×10^6 branch instructions. The CPI for each type of instruction is 1, 1, 4, and 2, respectively. Assume that the processor has a 2 GHz clock rate.

a. By how much must we improve the CPI of FP instructions if we want the program to run two times faster?

b. By how much must we improve the CPI of L/S instructions if we want the program to run two times faster?

c. By how much is the execution time of the program improved if the CPI of INT and FP instructions is reduced by 50% and the CPI of L/S and Branch is reduced by 25%?

2. When a program is adapted to run on multiple processors in a multiprocessor system, the execution time on each processor is comprised of computing time and the overhead time required for locked critical sections and/or to send data from one processor to another.

Assume a program requires t = 200 s of execution time on one processor. When running p processors, each processor requires t/p s, as well as an additional 10 s of overhead, irrespective of the number of processors. Compute the per-processor execution time for 2, 4, 8, 16, 32, 64 processors. For each case, list the corresponding speedup relative to a single processor and the ratio between actual speedup versus ideal speedup (speedup if there was no overhead).

3. Server farms such as Google and Yahoo! provide enough compute capacity for the highest request rate of the day. Imagine that most of the time these servers operate at only 60% capacity. Assume further that the power does not scale linearly with the load; that is, when the servers are operating at 60% capacity, they consume 90% of maximum power. The servers could be turned off, but they would take too long to restart in response to more load. A new system has been proposed that allows for a quick restart but requires 20% of the maximum power while in this "barely alive" state.

- **a.** How much power savings would be achieved by turning off 60% of the servers?
- **b.** How much power savings would be achieved by placing 60% of the servers in the "barely alive" state?
- **c.** How much power savings would be achieved by reducing the voltage by 20% and frequency by 40%?
- **d.** How much power savings would be achieved by placing 30% of the servers in the "barely alive" state and 30% off?

4. In a server farm such as that used by Amazon or eBay, a single failure does not cause the entire system to crash. Instead, it will reduce the number of requests that can be satisfied at any one time.

- **a.** If a company has 10,000 computers, each with a MTTF of 35 days, and it experiences catastrophic failure only if 1/3 of the computers fail, what is the MTTF for the system?

- **b.** If it costs an extra $1000, per computer, to double the MTTF, would this be a good business decision? Show your work.

5. In this exercise, assume that we are considering enhancing a machine by adding vector hardware to it. When a computation is run in vector mode on the vector hardware, it is **15 times faster** than the normal mode of execution. We call the percentage of time that could be spent using vector mode the *percentage of vectorization.* Vectors are discussed in Chapter 4, but you don't need to know anything about how they work to answer this question!

- **a.** Draw a graph that plots the speedup as a percentage of the computation performed in vector mode. Label the y-axis "Net speedup" and label the x-axis "Percent vectorization."
- **b.** What percentage of vectorization is needed to achieve **a speedup of 3**?
- **c.** What percentage of the computation run time is spent in vector mode **if a speedup of 2** is achieved?
- **d.** What percentage of vectorization is needed to achieve **one-half the maximum speedup attainable from using vector mode**?

e. Suppose you have measured the percentage of *vectorization of the program to be 70%*. The hardware design group estimates it can speed up the vector hardware even more with significant additional investment. You wonder whether the compiler crew *could increase the percentage of vectorization, instead*. What percentage of vectorization would the compiler team need to achieve in order to equal **an addition 2× speedup in the vector unit** (beyond the initial 15×)?

6. Assume for arithmetic, load/store, and branch instructions, a processor has CPIs of 2, 10, and 5, respectively. Also assume that on a single processor, a program requires the execution of 2.56E9 arithmetic instructions, 1.28E9 load/store instructions, and 128 million branch instructions. Assume that each processor has a 2GHz clock frequency.

Assume that, as the program is parallelized to run over multiple cores, the number of arithmetic and load/store instructions per processor is divided by 0.7 × p (where p is the number of processors) but the number of branch instructions per processor remains the same.

a. Find the total execution time for this program on 1, 2, 4, and 8 processors, and show the relative speedup of the 2, 4, and 8 processors result relative to the single processor result.

b. To what should the CPI of load/store instructions be reduced in order for a single processor to match the performance of four processors using the original CPI values?

7. Suppose we developed a simpler processor that has 75% of the capacitive load of a more complex processor. Further, assume that it can adjust voltage so that it can reduce voltage by 20% compared to the complex processor, but this results in a 25% increase in frequency.

a. How much energy do we save through this change?
b. What is the impact on the dynamic power?

8. One challenge for architects is that the design created today will require several years of implementation, verification, and testing before appearing on the market. This means that the architect must project what the technology will be like several years in advance. Sometimes, this is difficult to do.

a. According to the trend in device scaling historically observed by Moore's Law, the number of transistors on a chip in 2025 should be how many times the number in 2015? [assume # Transistors double every 2 years]
b. The increase in performance once mirrored this trend. Had performance continued to climb at the same rate as in the 1990s, approximately what performance would chips have over the VAX-11/780 in 2025? [assume performance increases 52% every year]
c. What has limited the rate of growth of the clock rate, and what are architects doing with the extra transistors now to increase performance?

9. General-purpose processes are optimized for general-purpose computing. That is, they are optimized for behavior that is generally found across a large number of applications. However, once the domain is restricted somewhat, the behavior that is found across a large number of the target applications may be different from general-purpose applications. One such application is deep learning or neural networks. Deep learning can be applied to many different applications, but the fundamental building block of inference—using the learned information to make decisions—is the same across them all. Inference operations are largely parallel, so they are currently performed on graphics processing units, which are specialized more toward this type of computation, and not to inference in particular. In a quest for more performance per watt, Google has created a custom chip using tensor processing units to accelerate inference operations in deep learning.1 This approach can be used for speech recognition and image recognition, for example. This problem explores the trade-offs between this process, a general-purpose processor (Haswell E5-2699 v3) and a GPU (NVIDIA K80), in terms of performance and cooling. If heat is not removed from the computer efficiently, the fans will blow hot air back onto the computer, not cold air. Note: The differences are more than processor—on-chip memory and DRAM also come into play. Therefore statistics are at a system level, not a chip level.

a. If Google's data center spends 70% of its time on workload A and 30% of its time on workload B when running GPUs, what is the speedup of the TPU system over the GPU system?

b. Google's data center spends 70% of its time on workload A and 30% of its time on workload B when running GPUs, what percentage of Max IPS does it achieve for each of the three systems?

c. Building on (b), assuming that the power scales linearly from idle to busy power as IPS grows from 0% to 100%, what is the performance per watt of the TPU system over the GPU system?

d. If another data center spends 40% of its time on workload A, 10% of its time on workload B, and 50% of its time on workload C, what are the speedups of the GPU and TPU systems over the general-purpose system?

e. A cooling door for a rack cost $4000 and dissipates 14 kW (into the room; additional cost is required to get it out of the room). How many Haswell-, NVIDIA-, or Tensor-based servers can you cool with one cooling door, assuming TDP in Figures 1.27 and 1.28?

f. Typical server farms can dissipate a maximum of 200 W per square foot. Given that a server rack requires 11 square feet (including front and back clearance), how many servers from part (e) can be placed on a single rack, and how many cooling doors are required?

System	Chip	TDP	Idle power	Busy power
General-purpose	Haswell E5-2699 v3	504 W	159 W	455 W
Graphics processor	NVIDIA K80	1838 W	357 W	991 W
Custom ASIC	TPU	861 W	290 W	384 W

Figure 1.27 Hardware characteristics for general-purpose processor, graphical processing unit-based or custom ASIC-based system, including measured power

System	Chip	Throughput			% Max IPS		
		A	B	C	A	B	C
General-purpose	Haswell E5-2699 v3	5482	13,194	12,000	42%	100%	90%
Graphics processor	NVIDIA K80	13,461	36,465	15,000	37%	100%	40%
Custom ASIC	TPU	225,000	280,000	2000	80%	100%	1%

Figure 1.28 Performance characteristics for general-purpose processor, graphical processing unit-based or custom ASIC-based system on two neural-net workloads

10. Consider the following two processors. P1 has a clock rate of 4GHz, average CPI of 0.9, and requires the execution of 5.0E9 instructions. P2 has a clock rate of 3GHz, an average CPI of 0.75, and requires the execution of 1.0E9 instructions.

a. One usual fallacy is to consider the computer with the largest clock rate as having the highest performance. Check if this is true for P1 and P2.

b. Another fallacy is to consider that the processor executing the largest number of instructions will need a larger CPU time. Considering that processor P1 is executing a sequence of 1.0E9 instructions and that the CPI of processors P1 and P2 do not change, determine the number of instructions that P2 can execute in the same time that P1 needs to execute 1.0E9 instructions.

c. A common fallacy is to use MIPS (millions of instructions per second) to compare the performance of two different processors, and consider that the processor with the largest MIPS has the largest performance. Check if this is true for P1 and P2.

11. A program runs in 100 seconds on a single-core processor. A new processor improves the performance of a specific task within the program by a factor of 5, but this task only accounts for 30% of the total execution time. Calculate the speedup achieved on the new processor using Amdahl's Law.