Sam Hutchins

Homework 1

Question 1 (6 points)

A) A technology metric of interest doubles every 15 months. How long will it take to improve 16x?

$$\text{Growth Factor} = 2^{\frac{t}{T}}$$
$$\text{Growth Factor} = 16$$
$$T = 15$$
$$t = ?$$
$$\log_2(16) = \frac{t}{15}$$
$$4 = \frac{t}{15}$$
$$t = 4 * 15 = 60 \text{ months}$$

B) For each of the scenarios below, decide whether the user will be most interested in latency, throughput, or both. Explain your reasoning for each

i) A cost-conscious cloud vendor picking a server to run interactive web requests

A cost-conscious cloud vendor picking a server to run interactive web requests should care about latency as it allows the user to to have faster times executing requests, which improves the user's experience, and throughput would only increase the total requests possible to handle at a time, which is less important to the user's perceived speed of the interactive web page.

ii) A user choosing which wireless provider to contract based on perceived speed

A user choosing which wireless provider to contract based on perceived speed should be interested in both latency and throughput as both will assist in a greater perceived speed when considering an internet connection. Latency will assist in quicker web searches, while throughput will assist in greater download speeds.

iii) A video editor choosing a workstation to help them create and process films

A video editor choosing a workstation to help them create and process films should care about throughput the most as it will assist in downloading large files, and processing large amounts of data such as 4k film. Responsiveness is not the priority here, but the handling of large amounts of data.

C) Assume you are manufacturing widgets. If you make 6 widgets at a time in a batch, and sustain a production rate of 36 widgets / hour, how long does it take to make 1 widget?

$$\text{Parallelism} = \text{Throughput} * \text{Latency}$$

$$\text{Throughput} = 36\frac{\text{widgets}}{\text{hour}}$$

$$\text{Parallelism} = 6 \text{ widgets}$$

$$\text{Latency} = ?$$

$$6 = 36 * \text{Latency}$$

$$\text{Latency} = \frac{6}{36} = 0.6 \text{ hours}$$

Question 2 (10 points)

You are designing a mobile device which has a battery capacity of 12 Wh.

A) Initially, the device has only one power mode (constant peak). That power mode consumes 6W. How long will the battery last?

$$\text{battery life} = \frac{12\text{Wh}}{6W}$$

$$\text{battery life} = 2 \text{ hours}$$

At constant peak, the battery will last 2 hours

B) To improve battery life, you add a low-power sleep mode that consumes only 2W. If the low-power mode is used half of the time, how long will the battery last?

$$P_{\text{avg}} = (t_{\text{low}} * P_{\text{low}}) + (t_{\text{normal}} * P_{\text{normal}})$$

$$t = .50$$

$$P_{\text{avg}} = (0.5 * 2) + (0.5 * 6)$$

$$P_{\text{avg}} = 1 + 3 = 4W$$

$$\text{battery life} = \frac{\text{battery capacity}}{P_{\text{avg}}}$$

$$\text{batter life} = \frac{12}{4} = 3 \text{ hours}$$

The battery will last 3 hours when low-power mode is used half of the time

C) To make the device last 5 hours, what fraction of the time must it spend in the low- power mode (from Part B)?

$$P_{\text{avg}} = \frac{\text{battery capacity}}{\text{battery life}}$$

$$P_{\text{avg}} = \frac{12}{5} = 2.4W$$

$$P_{\text{avg}} = (t_{\text{low}} * P_{\text{low}}) + (t_{\text{normal}} * P_{\text{normal}})$$

$$2.4 = (t_{\text{low}} * 2) + ((1 - t_{\text{low}}) * 6)$$

$$2.4 = 2t_{\text{low}} + 6 - 6t_{\text{low}}$$

$$2.4 = 6 - 4t_{\text{low}}$$

$$3.6 = 4t_{\text{low}}$$

$$t_{\text{low}} = \frac{3.6}{4} = 0.9$$

The device must spend 90% of time in low-power mode to last for 5 hours

Question 3 (10 points)

To improve your processor's performance, you consider adding a coprocessor to accelerate linear algebra. On linear algebra, the coprocessor obtains a 12x speedup. The benchmark workload is 60% linear algebra.

A) What overall speedup will your system with the coprocessor obtain on the benchmark?

$$\text{Speedup} = \frac{\text{CPUTime}_{\text{old}}}{\text{CPUTime}_{\text{new}}}$$

$$\text{Speedup} = \frac{\text{CPUTime}_{\text{old}}}{\text{CPUTime}_{\text{old}}\left[(1 - f_x) + \frac{f_x}{S_x}\right]}$$

$$\text{Speedup} = \frac{1}{(1 - f_x) + \frac{f_x}{S_x}}$$

$$\text{Speedup} = \frac{1}{(1 - 0.6) + \frac{0.6}{12}} = \sim 2.22x$$

The overall speedup will be approximately 2.22x

B) What is the maximum overall speedup possible for the benchmark by accelerating only linear algebra?

$$\text{Speedup}_{\text{max}} = \frac{1}{1 - f_x}$$

$$\text{Speedup}_{\text{max}} = \frac{1}{1 - 0.6} = \frac{1}{0.4} = 2.5x$$

The maximum overall speedup possible for the benchmark by accelerating only linear algebra is 2.5x

C) Working with your team, you come up with two ways to further improve your processor's performance after you added the coprocessor from part A. Unfortunately, you only have the resources to pursue one of them. Which option would you choose to maximize performance on the benchmark and why?

• Option A - Improve the performance of the processor on the rest of the workload (not linear algebra) by 1.2x.

• Option B - Double the number of operations the coprocessor can have in flight at a time. Each operation will still have the same latency.

Option A Analysis:

$$\text{unoptimized time} = \frac{1 - f_x}{1.2} = \frac{0.4}{1.2} = 0.333 \text{ Speedup} = \frac{1}{(\text{unoptimized time}) + \frac{f_x}{S_x}}$$

$$\text{Speedup} = \frac{1}{0.333 + \frac{0.6}{12}} = \frac{1}{0.333 + 0.05} = 2.61x$$

Option B Analysis:

double number of operations 12*2=24x speedup

$$\text{Speedup} = \frac{1}{(1 - f_x) + \frac{f_x}{S_x}}$$

$$\text{Speedup} = \frac{1}{(1 - 0.6) + \frac{0.6}{24}}$$

$$\text{Speedup} = \frac{1}{0.4 + 0.025} = \frac{1}{0.425} = 2.35x$$

Option A will result in a greater speedup of 2.65x, so it would be best to invest in option A first.

Question 4 (14 points)

A) A processor takes 8 seconds to execute a program and its clock rate is 2 GHz. If its CPI is 1.6, how many instructions are executed by the program?

$$\text{Clock Rate} = 2 \text{ GHz} = 2 * 10^9$$
$$\text{CPU Time} = 8 \text{ seconds}$$
$$\text{CPI} = 1.6$$
$$\text{Instruction Count} = \frac{\text{CPU Time} * \text{Clock Rate}}{\text{CPI}}$$
$$\text{Instruction Count} = \frac{8 * (2 * 10^9)}{1.6}$$
$$\text{Instruction Count} = \frac{16 * 10^9}{1.6}$$
$$\text{Instruction Count} = 10 * 10^9 = 10 \text{ billion instructions}$$

10 billion instructions are executed by the program

B) Processors A & B execute the same program P, and on both systems, it takes the same number of instructions. Processor B executes P 1.2x faster than Processor A. If Processor A's clock frequency is 1.6x faster than Processor B, how many times better is Processor B's IPC (not CPI)?

$$\text{freq}_A = \text{freq}_B * 1.6$$

$$\text{Execution Time of B} = \frac{\text{Execution Time of A}}{1.2}$$

$$\text{IPC} = \frac{\text{Instruction Count}}{\text{Clock Cycles}}$$

$$\text{IPC} = \frac{\text{Instruction Count}}{\text{Clock Rate} * \text{Execution Time}}$$

$$\text{IPC}_B = \frac{\text{Instruction Count}}{(\text{Clock Rate} * 1.6) * \text{Execution Time}}$$

$$\text{IPC}_A = \frac{\text{Instruction Count}}{\text{Clock Rate} * \frac{\text{Execution Time}}{1.2}}$$

$$\frac{\text{IPC}_B}{\text{IPC}_A} = \frac{(\text{Clock Rate} * 1.6) * \text{Execution Time}}{\text{Clock Rate} * \frac{\text{Execution Time}}{1.2}}$$

$$\frac{\text{IPC}_B}{\text{IPC}_A} = \frac{1.6}{\frac{1}{1.2}}$$

$$\text{IPC}_B = 1.92$$

C) Given a processor and workload with the following instruction mix and CPI:

i) What is the average CPI?

$$\text{Average CPI} = (0.4 * 3) + (0.10 * 4) + (0.3 * 6) + (0.2 * 7)$$

$$\text{Average CPI} = (1.2) + (0.4) + (1.8) + (1.4) = 4.8$$

The average CPI is 4.8

ii) With extensive engineering effort, you can reduce the time an instruction type takes by 1 cycle. Which instruction type will improve performance by the most? Why?

Reduction of Integer Arithmetic will improve the performance the most as it provides the most into the average CPI mix: 0.4 is the greatest

iii) You decide to add a store buffer, which will greatly speed up stores but will slow down loads. If the store buffer reduces the store CPI to 1, for this optimization to be worthwhile, what is the greatest CPI loads could be increased to? Ignore part ii.

$$\text{new average CPI} = (0.4 * 3) + (0.10 * 4) + (0.3 * \text{new CPI}) + (0.2 * 1)$$

$$4.8 > (0.4 * 3) + (0.10 * 4) + (0.3 * \text{New Load CPI}) + (0.2 * 1)$$

$$4.8 > 1.2 + 0.4 + (0.3 * \text{New Load CPI}) + 0.2$$

$$1.8 + (0.3 * \text{New Load CPI}) < 4.8$$

$$0.3 * \text{New Load CPI} < 3$$

$$\text{New Load CPI} < 10$$

The Load CPI can go up to a maximum of 10 to make store optimization worthwhile

iv) Consider the processor before any changes from parts ii and iii. You tweak the compiler to reduce the overall number of instructions executed by 0.9x, however, it causes half of the integer operations to

become stores (i.e. integer now 20% and stores now 40%). Will this optimization lead to a net speedup for this workload? Why?

- Integer Arithmetic decreased to 20%
- Stores increased to 40%

$$\text{New Average CPI} = (0.2 \times 3) + (0.1 \times 4) + (0.3 \times 6) + (0.4 \times 7)$$
$$\text{New Average CPI} = 0.6 + 0.4 + 1.8 + 2.8 = 5.6$$

Given the new average CPI and an instruction reduction of 0.9x, you get a net speed of 5.6 x 0.9 = 5.04x. This is greater than the original 4.8, so there is not an overall speedup on the system.