DEVIN MAYA 1/26/23

CSE 120 Homework 1

Question 1

A) A technology metric of interest doubles (2x) every 15 months starting from the base year. How long (in months) will it take to improve 1024x from the base year?

Solution. Given: Technology metric doubles every 15 months.

Objective: Find time to improve by 1024x from the base year.

Since the metric doubles every 15 months, this can be modeled as an exponential growth.

Let T be the total time in months for a 1024x improvement.

We know that 1024x improvement is equivalent to 2^{10} times.

Therefore, 2^{10} doubling periods are needed.

Each doubling period is 15 months. Thus,

T = 10 $\times$15 months

$\boxed{T = 150 \text{ months}}$

B) Assume you are manufacturing cars. If you make 15 cars in a batch, and sustain a production rate of 60 cars/hour, how long (in minutes or hours) does it take to make 1 car? Assume complete spatial parallelism between the 15 cars being made in the batch

Solution. In a car manufacturing process, where 15 cars are made in a batch and the production rate is 60 cars/hour, the time per car is calculated as:

$$\text{Time per car} = \frac{\frac{1}{4}\text{ hour}}{15} = \frac{1}{60}\text{ hour} = 1\text{ minute}$$

Given:

- 15 cars are made in a batch.
- Production rate = 60 cars/hour.

Objective: Find the time taken to make 1 car.

Assumption: Complete spatial parallelism, implying all 15 cars are made simultaneously.

Total time to produce one batch (15 cars) = $\frac{\text{Number of cars in a batch}}{\text{Production rate}}$

$= \frac{15\text{ cars}}{60\text{ cars/hour}} = \frac{1}{4}$ hour = 15 minutes

Since all 15 cars are made simultaneously, time taken to produce 1 car is the same as the batch.

Therefore, time taken to make 1 car = $\frac{1}{4}$ hour = 15 minutes

$\boxed{\text{Time per car} = 15 \text{ minutes}}$

Question 2

To improve your processor's performance on a certain program, you consider adding a coprocessor to accelerate the part of the program involved in Machine Learning. On machine learning alone, the coprocessor obtains a 8x speedup. The program comprises 60 percent machine learning.

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

Solution. To determine the overall speedup with a coprocessor accelerating 60% of the program 8 times, we use Amdahl's Law:

$$\text{Speedup} = \frac{1}{(1-0.6)+\frac{0.6}{8}}$$

$$\text{Speedup} = \frac{1}{0.4+0.075} = \frac{1}{0.475}$$

Objective: Calculate the overall speedup of the system with a coprocessor on the program.
Given:

- Coprocessor achieves an 8x speedup on the machine learning part.
- Machine learning constitutes 60% of the program.

Using Amdahl's Law, the speedup (S) can be calculated as:
Let f be the fraction of the program that benefits from the speedup (60%).
Let $\mathrm{S_{ml}}$ be the speedup on the machine learning part (8x).
Amdahl's Law formula: $\mathrm{S} = 1\frac{}{(1-f)+\frac{f}{S_{\mathrm{ml}}}}$
Substituting the values:
$\mathrm{S} = 1\frac{}{(1-0.6)+\frac{0.6}{8}}$
$= 1\frac{}{0.4+0.075}$
Therefore, the overall speedup is:

$$\boxed{S = \frac{1}{0.475} \approx 2.105}$$

B) What is the maximum overall speedup possible for the benchmark by accelerating only machine learning?

Solution. The goal is to determine the maximum overall speedup possible by only accelerating the machine learning part of a program.

Given: The acceleration applies only to the machine learning part.

We use Amdahl's Law to calculate the maximum speedup, denoted as S_{max}.

Let f represent the fraction of the program that can be accelerated, which in this case is 60% or 0.6.

As the speedup on the machine learning part approaches infinity, the formula for maximum speedup according to Amdahl's Law simplifies to:

$$S_{\max} = \frac{1}{1-f}$$

Substituting the value of $f = 0.6$, we get:

$$S_{\max} = \frac{1}{1-0.6} = \frac{1}{0.4}$$

Therefore, the maximum theoretical speedup achievable is:

$$\boxed{S_{\max} = 2.5}$$

Question 3

A) A processor takes 20 seconds to execute a program, and its clock rate is 4GHz. If its CPI is 5, how many instructions are executed by the program?

Solution. A processor takes 20 seconds to execute a program, and its clock rate is 4GHz. If its CPI is 5, how many instructions are executed by the program?

Solution: The total number of instructions executed can be calculated using the formula:

$$\text{Instruction count} = \frac{\text{Total number of cycles}}{\text{CPI}}$$

Given:

- Clock Rate = 4 GHz = 4×10^9 cycles/second
- Execution Time = 20 seconds
- CPI = 5

Total number of cycles = Clock Rate × Execution Time

$$= 4 \times 10^9 \times 20$$

Instruction count = $\frac{\text{Total number of cycles}}{\text{CPI}}$

$$= \frac{4 \times 10^9 \times 20}{5}$$

$$= 16 \times 10^9$$

Therefore, the number of instructions executed by the program is:

$$\boxed{16 \times 10^9 \text{ instructions}}$$

B) Processors A & B execute the same program P, and on both systems, it takes the same number of instructions. Processor A executes program P 1.5x slower than Processor B. If Processor A's clock frequency is 1.6x faster than Processor B, how many times better is Processor B's IPC (1/CPI)?

Solution. Processors A and B execute the same program P, and on both systems, it takes the same number of instructions. Processor A executes program P 1.5x slower than Processor B. If Processor A's clock frequency is 1.6x faster than Processor B, how many times better is Processor B's IPC (1/CPI)?

Solution: The IPC (Instructions Per Cycle) ratio can be calculated using the given information.

Given:

- Processor A is 1.5x slower than Processor B.
- Processor A's clock frequency is 1.6x faster than Processor B.

The IPC ratio between Processor B and Processor A is:

$$\frac{\text{IPC}_B}{\text{IPC}_A} = \frac{\text{Speed of Processor A}}{\text{Speed of Processor B}} \times \frac{\text{Clock frequency of Processor B}}{\text{Clock frequency of Processor A}}$$

$$= 1.5 \times \frac{1}{1.6}$$

C) Consider two different implementations, P1 and P2, of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (classes A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 4 respectively, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2 respectively. Given a program with an instruction count of 1.0E9 (this is the scientific notation for writing 1*10 9) instructions, is divided into classes as follows: 20and 30

a. Which implementation is faster to run the given program: P1 or P2?

Solution. To determine the faster implementation between P1 and P2, we compare their execution times. The execution time is given by the formula:

$$\text{Execution time} = \frac{\text{Instruction count} \times \text{Average CPI}}{\text{Clock rate}}$$

Given:

- Instruction count $= 1.0 \times 10^9$ instructions.
- Distribution: 20% class A, 10% class B, 40% class C, 30% class D.

- P1: Clock rate = 2.5 GHz, CPIs = 1, 2, 3, 4 for classes A, B, C, D respectively.
- P2: Clock rate = 3 GHz, CPIs = 2 for all classes.

For P1:

$$\text{Average CPI}_{P1} = (0.20 \times 1) + (0.10 \times 2) + (0.40 \times 3) + (0.30 \times 4) = 2.5$$

$$\text{Execution time}_{P1} = \frac{1.0 \times 10^9 \times 2.5}{2.5 \times 10^9} = 1 \text{ second}$$

For P2:

$$\text{Average CPI}_{P2} = 2$$

$$\text{Execution time}_{P2} = \frac{1.0 \times 10^9 \times 2}{3 \times 10^9} = 0.67 \text{ seconds}$$

Since Execution time$_{P2}$ < Execution time$_{P1}$, P2 is faster.

b. What is the global (average) CPI for each implementation?

Solution. The global CPI for each implementation is calculated based on the distribution of instruction classes and their respective CPIs.

For P1:

$$\text{Average CPI}_{P1} = (0.20 \times 1) + (0.10 \times 2) + (0.40 \times 3) + (0.30 \times 4) = 2.5$$

For P2:

$$\text{Average CPI}_{P2} = 2 \text{ (since all classes have a CPI of 2)}$$

c. Find the number of clock cycles required to execute the program in both cases.

Solution. The number of clock cycles required is calculated using the formula:

$$\text{Clock cycles} = \text{Instruction count} \times \text{Average CPI}$$

For P1:

$$\text{Clock cycles}_{P1} = 1.0 \times 10^9 \times 2.5 = 2.5 \times 10^9 \text{ cycles}$$

For P2:

$$\text{Clock cycles}_{P2} = 1.0 \times 10^9 \times 2 = 2.0 \times 10^9 \text{ cycles}$$

Question 4 (10 points)

In this problem, we are going to study how adding an increasing number of cores in a practical system modifies its speedup. Assume a single core has the following CPIs for each class of instructions: 1. Arithmetic= 2 2. Load/Store(L/S)= 15 3. Branch=5 Each core runs on a 2GHz clock frequency Also assume that we are executing a program in our system comprising the following number of instructions: 1. Arithmetic=2.56E9 (this is the same as writing 2.56 * 10 9) 2.

Load/Store(L/S)=1.28E9 3. Branch=2.56E8 Now, assume that, as the program is parallelized to run over multiple cores, the number of arithmetic and load/store instructions per core is divided by (0.9 × p), where p is the number of cores, but the number of branch instructions per processor remains the same. So, for example, if p=2, we divide the arithmetic and L/S instructions per core by 0.9*2=1.8 each but keep branch instructions the same. (Ideally, we would divide the above parallelable instructions just by p; but sometimes in real life, even the same class of instructions cannot be fully parallelized, hence the divide by 0.9 * p instead in this example)

A) Find the total execution time for this program on p= 1, 2 and 4 cores, and show the relative speedup of the 2 and 4 cores result relative to the single core result. Accordingly, fill up the values in the table provided below (5pts)

Solution. Calculate the total execution time for this program on $p = 1, 2$, and 4 cores, and show the relative speedup of the 2 and 4 cores result relative to the single core result.

Given:

- Clock frequency = 2×10^9 cycles/second.
- CPIs: Arithmetic = 2, Load/Store = 15, Branch = 5.
- Instruction counts: Arithmetic = 2.56×10^9, Load/Store = 1.28×10^9, Branch = 2.56×10^8.
- For multiple cores, arithmetic and L/S instructions per core are divided by $0.9 \times p$, while branch instructions remain the same.

Calculations:

For $p = 1$:

$$\text{Total cycles} = (2 \times 2.56 \times 10^9) + (15 \times 1.28 \times 10^9) + (5 \times 2.56 \times 10^8) = 28,302,222,222 \text{ cycles}$$

$$\text{Execution time} = \frac{28,302,222,222}{2 \times 10^9} \text{ seconds} = 14.15 \text{ seconds}$$

For $p = 2$:

$$\text{Arithmetic and L/S per core} = \frac{\text{Original count}}{0.9 \times 2}$$

$$\text{Total cycles for 2 cores} = 14,791,111,111 \text{ cycles}$$

$$\text{Execution time} = 7.40 \text{ seconds}$$

For $p = 4$:

$$\text{Arithmetic and L/S per core} = \frac{\text{Original count}}{0.9 \times 4}$$

$$\text{Total cycles for 4 cores} = 8,035,555,556 \text{ cycles}$$

$$\text{Execution time} = 4.02 \text{ seconds}$$

Speedup Calculation:

$$\text{Speedup for } p = 2 = \frac{14.15}{7.40} = 1.91$$

$$\text{Speedup for } p = 4 = \frac{14.15}{4.02} = 3.52$$

p	#arith inst.	#L/S inst.	#branch inst.	cycles	ex. time (s)	speedup
1	2.56E9	1.28E9	2.56E8	28,302,222,222	14.15	1
2	1.42E9	0.71E9	2.56E8	14,791,111,111	7.40	1.91
4	0.71E9	0.36E9	2.56E8	8,035,555,556	4.02	3.52

B) If the CPI of ONLY the Load/Store instructions was doubled, what would the impact be on the execution time of the program on p= 1, 2, 4 processors?

Solution. If the CPI of ONLY the Load/Store instructions was doubled, what would be the impact on the execution time of the program on $p = 1, 2, 4$ processors?

Given:

- New CPI for Load/Store = 30 (doubled from 15).
- Arithmetic CPI = 2, Branch CPI = 5.
- Instruction counts: Arithmetic = 2.56×10^9, Load/Store = 1.28×10^9, Branch = 2.56×10^8.
- For multiple cores, arithmetic and L/S instructions per core are divided by $0.9 \times p$, while branch instructions remain the same.

Calculations:

For $p = 1$:

$$\text{Total cycles} = (2 \times 2.56 \times 10^9) + (30 \times 1.28 \times 10^9) + (5 \times 2.56 \times 10^8) = 51,712,222,222 \text{ cycles}$$

$$\text{Execution time} = \frac{51,712,222,222}{2 \times 10^9} \text{ seconds} = 25.86 \text{ seconds}$$

For $p = 2$:

$$\text{Arithmetic and L/S per core} = \frac{\text{Original count}}{0.9 \times 2}$$

$$\text{Total cycles per core} = \text{Similar calculation as for } p = 1$$

$$\text{Total cycles for 2 cores} = 26,400,000,000 \text{ cycles}$$

$$\text{Execution time} = 13.20 \text{ seconds}$$

For $p = 4$:

$$\text{Arithmetic and L/S per core} = \frac{\text{Original count}}{0.9 \times 4}$$

$$\text{Total cycles per core} = \text{Similar calculation as for } p = 1$$

$$\text{Total cycles for 4 cores} = 13,835,555,556 \text{ cycles}$$

$$\text{Execution time} = 6.92 \text{ seconds}$$

p	cycles	ex. time (s)
1	51,712,222,222	25.86
2	26,400,000,000	13.20
4	13,835,555,556	6.92

C) What should the CPI of load/store instructions be reduced to in order for a modified single-core processor to match the performance of a four-core processor using the original CPI values? Hint: Equate the two different execution times and solve for the load/store CPI

Solution. To find the required CPI for Load/Store instructions for a single-core processor to match the performance of a four-core processor using the original CPI values, we equate the execution times and solve for the new CPI.

Given:

- Execution time of a four-core processor with original CPI values = 4.02 seconds.
- Arithmetic CPI = 2, Branch CPI = 5, and original Load/Store CPI = 15.
- Instruction counts: Arithmetic = 2.56×10^9, Load/Store = 1.28×10^9, Branch = 2.56×10^8.

The execution time for the single-core processor with the modified CPI for L/S instructions should be 4.02 seconds.

Calculations:

For the single-core processor:

$$\text{Total cycles} = (2 \times 2.56 \times 10^9) + (CPI_{L/S} \times 1.28 \times 10^9) + (5 \times 2.56 \times 10^8)$$

Setting the execution time equal to 4.02 seconds:

$$4.02 = \frac{(2 \times 2.56 \times 10^9) + (CPI_{L/S} \times 1.28 \times 10^9) + (5 \times 2.56 \times 10^8)}{2 \times 10^9}$$

Solving for $CPI_{L/S}$:

$$CPI_{L/S} = \frac{4.02 \times 2 \times 10^9 - (2 \times 2.56 \times 10^9) - (5 \times 2.56 \times 10^8)}{1.28 \times 10^9}$$

$$CPI_{L/S} \approx 7.4$$

Therefore, the required CPI for Load/Store instructions for a single-core processor to match the performance of a four-core processor is approximately 7.4.