

Assignment-1: Performance Metrics

Arnab Das, u1014840

January 30, 2018

1 Summarizing Performance Numbers

Execution Time: The expected execution time for each machine can be derived as the arithmetic mean of the individual execution times since the probability of executing each program is equal. Arithmetic mean is defined as

$$ArithMean = \frac{\sum_{i=1}^{n=4} x_i}{n}$$

The evaluation is summarised in 1

| Execution Times(S) | | | | | |
|--------------------|---|-----|---|----|-------|
| System | A | B | C | D | WSum |
| BASE | 3 | 2.5 | 1 | 12 | 4.625 |
| NEW1 | 7 | 3 | 5 | 1 | 4 |
| NEW2 | 2 | 1 | 3 | 8 | 3.5 |
| NEW3 | 1 | 3 | 2 | 13 | 4.75 |

Table 1: execution times

Thus, the computer system resulting in least execution time is **NEW2**.

Energy: Similarly, for the least energy consumption we evaluate the arithmetic mean of the energy values over all the programs for each system. The evaluated arithmetic mean for energy table is shown in 2

The evaluations are shown in the table 2.

Thus the system **NEW1** has the least energy consumption.

Power: From the definition of power as

$$Power = \frac{Energy}{Time}$$

| System Energy(J) | | | | | |
|------------------|----|----|----|----|-----------|
| System | A | B | C | D | ArithMean |
| BASE | 20 | 40 | 50 | 15 | 31.25 |
| NEW1 | 10 | 30 | 15 | 30 | 21.25 |
| NEW2 | 30 | 60 | 20 | 20 | 32.5 |
| NEW3 | 70 | 35 | 30 | 10 | 36.25 |

Table 2: Energy Consumption

we can construct the table for *power* consumption using the corresponding values of energy and execution-times from the given tables using the above equation. Table 3 shows the above construction

| System Power(W) | | | | | |
|-----------------|-----------------------|------------------------|-----------------------|------------------------|-----------|
| System | A | B | C | D | AirthMean |
| BASE | $\frac{20}{3} = 6.67$ | $\frac{40}{2.5} = 16$ | $\frac{50}{1} = 50$ | $\frac{15}{12} = 1.25$ | 18.48 |
| NEW1 | $\frac{10}{7} = 1.43$ | $\frac{30}{3} = 10$ | $\frac{15}{5} = 3$ | $\frac{30}{1} = 30$ | 11.12 |
| NEW2 | $\frac{30}{2} = 15$ | $\frac{60}{1} = 60$ | $\frac{20}{3} = 6.67$ | $\frac{20}{8} = 2.5$ | 21.04 |
| NEW3 | $\frac{70}{1} = 70$ | $\frac{35}{3} = 11.67$ | $\frac{30}{2} = 15$ | $\frac{10}{13} = 0.77$ | 24.36 |

Table 3: System Power

2 Optimizing CPU Time

Let the total instruction count be x . Thus we have the following count of instructions per type of instructions

$$\begin{aligned} \text{Load} &= 0.1x, \text{ and, } \# \text{ cycles} = 0.2x \\ \text{Store} &= 0.05x, \text{ and, } \# \text{ cycles} = 0.05x \\ \text{Branch} &= 0.05x, \text{ and, } \# \text{ cycles} = 0.1x \\ \text{ADD} &= 0.3x, \text{ and, } \# \text{ cycles} = 0.3x \\ \text{MULT} &= 0.5x, \text{ and, } \# \text{ cycles} = 2x \end{aligned}$$

Thus the Old CPI will be

$$CPI_{old} = \frac{0.2x + 0.05x + 0.1x + 0.3x + 2x}{x} = 2.65$$

Hence, the old IPC will be

$$IPC_{old} = \frac{1}{2.65} = 0.377$$

Given that 60% of the MULT instructions could be combined with ADD, we observe that 60% of the MULT instructions is $0.3x$. Thus all the ADD instructions get combined with the MULT instructions, leaving $0.2x$ MULT instructions that are unfused. Thus, the new bunch of instructions and their cycle count will be as follows

$$\begin{aligned} \text{Load} &= 0.1x, \text{ and, } \# \text{ cycles} = 0.2x \\ \text{Store} &= 0.05x, \text{ and, } \# \text{ cycles} = 0.05x \\ \text{Branch} &= 0.05x, \text{ and, } \# \text{ cycles} = 0.1x \\ \text{MULT} &= 0.2x, \text{ and, } \# \text{ cycles} = 0.8x \\ \text{FMAD} &= 0.3x, \text{ and, } \# \text{ cycles} = 1.2x \end{aligned}$$

Thus the New CPI will be

$$CPI_{new} = \frac{0.2x + 0.05x + 0.1x + 0.8x + 1.2x}{0.1x + 0.05x + 0.05x + 0.2x + 0.3x} = 3.35$$

Hence, the new IPC will be

$$IPC_{new} = \frac{1}{3.35} = 0.29$$

Old execution time can be evaluated as

$$Exec_{old} = CPI_{old} \times IC_{old} \times CT = 2.65x \times CT$$

New execution time can be evaluated as

$$Exec_{new} = CPI_{new} \times IC_{new} \times CT = \frac{2.35x}{0.7x} \times 0.7x \times CT$$

Hence,

$$Speed - up = \frac{Exec_{old}}{Exec_{new}} = \frac{2.65x}{2.35x} = 1.127$$

Thus there is a net speedup of 1.127 using *FMAD*.

3 Amdahl's Law

Let the total energy consumption in the mobile device is x Joules. The fractions of energy consumed by each part of the given unit can be summarized in Table: 4

| Energy Distribution | | | |
|----------------------------|----------------------------|-----------------------------|----------------------------|
| Wireless Interface | Display | CPU | Other |
| $50\% \equiv \frac{1}{2}x$ | $20\% \equiv \frac{1}{5}x$ | $10\% \equiv \frac{1}{10}x$ | $20\% \equiv \frac{1}{5}x$ |

Table 4: System Power

We can apply Amdahl's law to find maximum improvement possible when some fraction of the component is improved. In this case the component is energy. We denote by $E_{overall}$ as the overall energy of the system and $f_{enhanced}$ as the fraction of the enhanced component. Then the formula for the specific case of this problem in terms of energy can be summarized as:

$$E_{overall_{new}} = (1 - f_{enhanced} + \frac{f_{enhanced}}{\text{Improvement speedup}}) \times E_{overall_{old}}$$

$$\text{Speedup} = \frac{E_{overall_{old}}}{E_{overall_{new}}}$$

Where, in case of energy, the *Improvement speedup* can be defined as

$$\text{Improvement speedup} = \frac{Energy_{enhanced_{old}}}{Energy_{enhanced_{new}}}$$

Applying the above set of formulae to the given cases for energy optimization

Reduce wireless interface energy by 10%

Then we get improvement speedup and the new Overall energy as

$$\begin{aligned}
 \text{Improvement Speedup} &= \frac{Energy_{enhanced_{old}}}{Energy_{enhanced_{new}}} = \frac{\frac{x}{2}}{\frac{x}{2} - 10\% \text{ of } \frac{x}{2}} \\
 &= \frac{10}{9} \\
 E_{overall_{new}} &= (1 - f_{enhanced} + \frac{f_{enhanced}}{\text{Improvement speedup}}) \times E_{overall_{old}} \\
 &= (1 - \frac{1}{2} + \frac{\frac{2}{10}}{\frac{10}{9}}) \times E_{overall_{old}} \\
 &= 0.95 E_{overall_{old}}
 \end{aligned}$$

Thus reducing wireless interface energy by 10% results in an **overall energy reduction of 5%**.

Reducing CPU energy by 60%

Then we get improvement speedup and the new Overall energy as

$$\begin{aligned}
 \text{Improvement Speedup} &= \frac{Energy_{enhanced_{old}}}{Energy_{enhanced_{new}}} = \frac{\frac{x}{10}}{\frac{x}{10} - 60\% \text{ of } \frac{x}{10}} \\
 &= \frac{5}{2} \\
 E_{overall_{new}} &= (1 - f_{enhanced} + \frac{f_{enhanced}}{\text{Improvement speedup}}) \times E_{overall_{old}} \\
 &= (1 - \frac{1}{10} + \frac{\frac{10}{5}}{\frac{5}{2}}) \times E_{overall_{old}} \\
 &= 0.94 E_{overall_{old}}
 \end{aligned}$$

Thus reducing CPU energy by 60% results in an **overall energy reduction of 6%**.

Reducing display energy by 50%

Then we get improvement speedup and the new Overall energy as

$$\begin{aligned}
 \text{Improvement Speedup} &= \frac{Energy_{enhanced_{old}}}{Energy_{enhanced_{new}}} = \frac{\frac{x}{10}}{\frac{x}{5} - 50\% \text{ of } \frac{x}{5}} \\
 &= 2 \\
 E_{overall_{new}} &= (1 - f_{enhanced} + \frac{f_{enhanced}}{\text{Improvement speedup}}) \times E_{overall_{old}} \\
 &= (1 - \frac{1}{5} + \frac{1}{2}) \times E_{overall_{old}} \\
 &= 0.90 E_{overall_{old}}
 \end{aligned}$$

Thus reducing display energy by 50% results in an **overall energy reduction of 10%**.

Thus, we see that the third option of reducing display energy by 50% gives the best improvement in energy saving since the overall energy gets reduced by 10%.

4 Power and Energy

Given,

$$\begin{aligned}\text{Frequency, } f &= 2\text{Ghz} \\ \text{Dynamic power, } P_d &= 70\text{W} \\ \text{Static power, } P_s &= 30\text{W} \\ T_{exec} &= 15\text{s}\end{aligned}$$

i. Compute the energy consumed for executing the application:

$$\begin{aligned}\text{Total Energy} &= (P_s + P_d) \times T_{exec} \\ &= (70 + 30) \times 15\text{Joules} \\ &= 1500\text{Joules}\end{aligned}$$

$\text{Total Energy} = 1500 \text{ Joules}$

ii. Energy consumption of frequency scales down by 30%:

Scaling down frequency affects inversely the executuin time as well, and the Dynamic power component. The new frequency and execution time will be

$$\begin{aligned}f_{new} &= 0.7f \\ T_{exec_{new}} &= \frac{T_{exec}}{0.7}\end{aligned}$$

The dynamic power componnet is expressed as

$$\text{Power}_{dyn} = \text{Activity} \times \text{Capacitance} \times \text{Voltage}^2 \times \text{Frequency}$$

In dynamic power, all the other components remaining same, the dynamic power scales by the same amount that frequency scales. Hence, the new dynamic power is

$$P_{d_{new}} = 0.7 \times P_d$$

Thus, the new energy consumption be comes:

$$\begin{aligned}\text{New Total Energy} &= (P_s + 0.7P_d) \frac{T_{exec}}{0.7} \\ &= (30 + 0.7 \times 70) \frac{15}{0.7} \\ &= 1692\end{aligned}$$

$\text{New Total Energy} = 1692 \text{ Joules}$

iii. *Energy consumption if both voltage and frequency scale down by 30%:*

Additionally if we scale the voltage down by 30%, it affects both dynamic power and static power as shown.

$$P_{static} = Current \times Voltage = 0.7 \times P_s$$

$$P_{dynamic} = Activity \times Capacitance \times Voltage^2 \times Frequency = (0.7)^3 \times P_d$$

$$New\ Total\ Energy = (0.7 \times P_s + (0.7)^3 \times P_d) \times \frac{15}{0.7} = 964.5\ Joules$$

| |
|--------------------------------------|
| $New\ Total\ Energy = 964.5\ Joules$ |
|--------------------------------------|

5 Instruction set Architecture

- **LOAD R5, 6000(R0)**

$$\text{Effective Address} = 6000 + [R0] = 6000 + 1000 = 7000$$

$$R5 \leftarrow \text{Mem}[7000]$$

$$R5 \leftarrow 1$$

Calculates the effective address as shown of 7000, then load the data from memory address 7000 into R5. R5 is updated to 1

- **ADD R4, (R4)**

$$\text{Effective Address} = \text{Mem}[R4] = 6000$$

$$R4 \leftarrow R4 + \text{Mem}[6000] = 6000 + 12 = 6012$$

- **SUB R2, R1**

$$R2 \leftarrow R2 - R1$$

$$R2 \leftarrow 99 - 25 = 74$$

- **LOAD R6, @(R0)**

$$\text{Effective Address} = \text{Mem}[R0] = \text{Mem}[1000] = 3000$$

$$R6 \leftarrow \text{Mem}[3000]$$

$$R6 \leftarrow 33$$

- **ADD R6, R4**

$$R6 \leftarrow R6 + R4$$

$$R6 \leftarrow 33 + 6012$$

$$R6 \leftarrow 6045$$

- **SUB R5, R6**

$$R5 \leftarrow R5 - R6$$

$$R5 \leftarrow 1 - 6045$$

$$R5 \leftarrow -6044$$

- **ADD R2, R5**

$$R2 \leftarrow R2 + R5$$

$$R2 \leftarrow 74 - 6044$$

$$R2 \leftarrow -5970$$

- **ADD R2, (R3+R0)**

$$\text{Effective Address} = R3 + R0 = 4000 + 1000 = 5000$$

$$R2 \leftarrow R2 + Mem[5000]$$

$$R2 \leftarrow -5970 + 71$$

$$R2 \leftarrow -5899$$