Partial Differential Equations: An Overview[1]

[1] CS 6220: Spring 2018: SoC, UoU

Arnab Das, u1014840

January 30, 2018

1 1. Summarizing Performance Numbers

Execution Time To find the system with the least execution time, we consider the execution times in the **BASE** system as reference to relativise how fast or how slow the benchmarks programs execute in the other systems. To do this, consider the execution time encountered in the **BASE** system as the reference for the corresponding benchmark program, and divide the execution numbers from the other system by the reference value. Hence, the reciprocal of the execution number in the **BASE** system serves as the weights as shown in Table **??**. The **weighted sum**(Wsum) is evaluated as:

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

Thus the system **NEW3** has the least execution time, considering they are equally probable to be executed. Note that the **Wsum** values do not have any unit and is a relative quantity with respect to that of **BASE**.

Table 1: Relativise execution times

Execution Times(S)					
System	**A**	**B**	**C**	**D**	**WSum**
BASE	$\frac{3}{3} = 1.00$	$\frac{2.5}{2.5} = 1.0$	$\frac{1}{1} = 1.0$	$\frac{12}{12} = 1.0$	1.0
NEW1	$\frac{7}{3} = 2.33$	$\frac{3}{2.5} = 1.2$	$\frac{5}{1} = 5.0$	$\frac{1}{12} = 0.08$	2.15
NEW2	$\frac{2}{3} = 0.67$	$\frac{1}{2.5} = 0.4$	$\frac{3}{1} = 3.0$	$\frac{8}{12} = 0.67$	1.1825
NEW3	$\frac{1}{3} = 0.33$	$\frac{3}{2.5} = 1.2$	$\frac{2}{1} = 2.0$	$\frac{13}{12} = 1.08$	1.15

Energy: Similarly, for the energy table we relativise the energy consumption values of the benchmark programs on the new system with respect to the **BASE** system by taking it as the reference. The evaluations are shown in the table **??**.

Table 2: Relativise Energy Values

System Energy(J)					
System	**A**	**B**	**C**	**D**	**WSum**
BASE	$\frac{20}{20} = 1.00$	$\frac{40}{40} = 1.00$	$\frac{50}{50} = 1.00$	$\frac{15}{15}$	1
NEW1	$\frac{10}{20} = 0.50$	$\frac{30}{40} = 0.75$	$\frac{15}{50} = 0.30$	$\frac{30}{15} = 2.00$	0.88
NEW2	$\frac{30}{20} = 1.50$	$\frac{60}{40} = 1.50$	$\frac{20}{50} = 0.40$	$\frac{20}{15} = 1.33$	1.18
NEW3	$\frac{70}{20} = 3.50$	$\frac{30}{40} = .875$	$\frac{30}{50} = 0.60$	$\frac{10}{15} = 0.67$	1.41

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

Power: From the defnition of power as

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

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 **??** shows the above construction

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

Table 3: System Power

2 1. *Optimizing CPU Time*

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

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

Thus the Old CPI will be

$$\text{CPI}_{old} = \frac{0.2x + 0.05x + 0.1x + 0.3x + 2x}{x} = 2.65$$

Hence, the old IPC will be

$$\text{IPC}_{old} = \frac{1}{2.65} = 0.377$$

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 instrcutions and their cycle count will be as follows

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

Thus the New CPI will be

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

Hence, the new IPC will be

$$\text{IPC}_{new} = \frac{1}{3.35} = 0.29$$

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 $FMAD$.