

Módulo 2 :: AC :: LEI

12 de Outubro 2012

Questão 1

a) $\#I_1 = ?$, $\#CC_1 = ?$, $CPI_1 = ?$, $\#I_2 = ?$, $\#CC_2 = ?$, $CPI_2 = ?$, $ganho = ?$

Compilador C1:

$$\#I_1 = \sum_{n=1}^3 \#I_1[n] = 8 * 10^6$$

$$\#CC_1 = \sum_{n=1}^3 \#I_1[n] * cpi[n] = (1 * 1 + 3 * 2 + 4 * 3) * 10^6 = 19 * 10^6$$

$$CPI_1 = \#CC_1 / \#I_1 = 19 / 8 = 2.375$$

Compilador C2:

$$\#I_2 = 10 * 10^6$$

$$\#CC_2 = (5 * 1 + 2 * 2 + 3 * 3) * 10^6 = 18 * 10^6$$

$$CPI_2 = 18 / 10 = 1.8$$

→ O C2 é mais rápido porque exige menos ciclos: $\#CC_2 < \#CC_1$.

$$ganho_{C2 \text{ sobre } C1} = \frac{Texe_1}{Texe_2} = \frac{\#CC_1 * T_{cpu}}{\#CC_2 * T_{cpu}} = \frac{\#CC_1}{\#CC_2}$$

$$ganho_{C2 \text{ sobre } C1} = \#CC_1 / \#CC_2 = 1.056 \rightarrow C2 \text{ é } 1.056 \text{ vezes mais rápido que } C1$$

b) $f_{cpu} = 1 \text{ GHz} = 10^9 \text{ Hz}$. $Texe_1 = ?$, $Texe_2 = ?$

$$Texe_1 = \#CC_1 * T_{cpu} = \#CC_1 / f_{cpu} = 19 * 10^6 / 10^9 = 19 * 10^{-3} = 19 \text{ ms}$$

$$Texe_2 = \#CC_2 * T_{cpu} = \#CC_2 / f_{cpu} = 18 * 10^6 / 10^9 = 18 * 10^{-3} = 18 \text{ ms}$$

c) $f_{cpu} = 2 \text{ GHz}$ e novos CPI's dados pela tabela 3. $CPI_1 = ?$, $Texe_1 = ?$, $CPI_2 = ?$, $Texe_2 = ?$

$$\#CC_1 = \sum_{n=1}^3 \#I_1[n] * cpi[n] = (1 * 2 + 3 * 3 + 4 * 4) * 10^6 = 27 * 10^6$$

$$CPI_1 = \#CC_1 / \#I_1 = 27 / 8 = 3.375$$

$$Texe_1 = \#CC_1 / f_{cpu} = 27 * 10^6 / 2 * 10^9 = 13.5 * 10^{-3} = 13.5 \text{ ms}$$

$$\#CC_2 = \sum_{n=1}^3 \#I_2[n] * cpi[n] = (5 * 2 + 2 * 3 + 3 * 4) * 10^6 = 28 * 10^6$$

$$CPI_2 = \#CC_2 / \#I_2 = 28 / 10 = 2.8$$

$$Texe_2 = \#CC_2 / f_{cpu} = 28 * 10^6 / 2 * 10^9 = 14 * 10^{-3} = 14 \text{ ms}$$

Questão 2

a) $CPI_{base} = ?$ e $CPI_{hard} = ?$

$base \rightarrow f_{cpu} = 1.5 \text{ GHz}$, CPI_{base} dado pela 2ª coluna da tabela 4

$hard \rightarrow f_{cpu} = 2 \text{ GHz}$, CPI_{hard} dado pela 3ª coluna da tabela 4

$$CPI_{base} = \sum_{n=1}^4 Freq[n] * cpi_{base}[n] = 0.4 * 2 + 0.25 * 3 + 0.25 * 3 + 0.1 * 5 = 2.8$$

$$CPI_{hard} = \sum_{n=1}^4 Freq[n] * cpi_{hard}[n] = 0.4 * 2 + 0.25 * 2 + 0.25 * 3 + 0.1 * 4 = 2.45$$

b) $MIPS_{base} = ?$, $MIPS_{hard} = ?$, $MIPS_{base_pico} = ?$ e $MIPS_{hard_pico} = ?$

$$Texe = \#I * CPI * T_{cpu} = \frac{\#I * CPI}{f_{cpu}}$$

$$MIPS = \frac{\#I}{Texe * 10^6} = \frac{\#I}{\frac{\#I * CPI}{f_{cpu}} * 10^6} = \frac{f_{cpu}}{CPI * 10^6} \quad (\text{Milhões de Instruções Por Segundo})$$

$$MIPS_{base} = 1.5 * 10^9 / (2.8 * 10^6) = 0,5357 * 10^3 = 536 \text{ MIPS}$$

$$MIPS_{hard} = 2 * 10^9 / (2.45 * 10^6) = 0,8163 * 10^3 = 816 \text{ MIPS}$$

Para obter o **MIPS de pico** considera-se o CPI mais favorável (mínimo): **2** (CPI das [instruções do tipo A](#))

Nota: um programa só com instruções do tipo A é “irreal”.

$$MIPS_{base_pico} = \frac{f_{cpu}}{CPI * 10^6} = 1.5 * 10^9 / (2 * 10^6) = 750 \text{ MIPS}$$

$$MIPS_{hard_pico} = \frac{2 * 10^9}{(2 * 10^6)} = 1000 \text{ MIPS}$$

c) $ganho_{hard \text{ sobre } base} = ?$

$$ganho_{hard \text{ sobre } base} = \frac{Texe_{base}}{Texe_{hard}} = \frac{(\#I_{base} * CPI_{base}) / f_{base}}{(\#I_{hard} * CPI_{hard}) / f_{hard}} = \frac{\#I_{base} * CPI_{base} * f_{hard}}{\#I_{hard} * CPI_{hard} * f_{base}}$$

Como se trata do mesmo programa: $\#I_{base} = \#I_{hard}$

$$ganho_{hard \text{ sobre } base} = \frac{CPI_{base} * f_{hard}}{CPI_{hard} * f_{base}} = \frac{2.8 * 2}{2.45 * 1.5} = 1.524$$

\rightarrow a opção **hard** é 1,524 vezes mais rápida que a opção **base**

d) Opção **comp**: opção base com novas percentagens de instruções geradas pelo compilador (**tabela 5**).

$CPI_{comp} = ?$, $ganho_{comp \text{ sobre } base} = ?$ e $ganho_{comp \text{ sobre } hard} = ?$

É necessário recalcular a frequência de cada tipo de instrução, não esquecendo de normalizar para 100% :

$$FreqA' = 0.4 * 0.9 = 0.36$$

$$FreqB' = 0.25 * 0.9 = 0.225$$

$$FreqC' = 0.25 * 0.85 = 0.2125$$

$$FreqD' = 0.1 * 0.95 = 0.095$$

$$FreqA' + FreqB' + FreqC' + FreqD' = 0.8925 \quad (89.25\% \text{ é menor que } 100\%, \text{ logo é necessário normalizar para } 100\%)$$

$$FreqA = 0.36 / 0.8925 = 0.4033$$

$$FreqB = 0.225 / 0.8925 = 0.2521$$

$$FreqC = 0.2125 / 0.8925 = 0.2381$$

$$FreqD = 0.095 / 0.8925 = 0.1064$$

$$CPI_{comp} = \sum_{n=1}^4 Freq_{comp}[n] * cpi_{base}[n] = 0.4033 * 2 + 0.2521 * 3 + 0.2381 * 3 + 0.1064 * 5 = 2.8094$$

Notas: #I_comp = 0.8925 * #I_base e T_cpu_base = T_cpu_comp (mesma frequência)

$$\begin{aligned} \text{ganho}_{comp \text{ sobre base}} &= \frac{Texe_{base}}{Texe_{comp}} = \frac{\#I_{base} * CPI_{base} * Tcpu_{base}}{\#I_{comp} * CPI_{comp} * Tcpu_{comp}} = \frac{\#I_{base} * CPI_{base} * Tcpu_{base}}{0.8925 * \#I_{base} * CPI_{comp} * Tcpu_{comp}} = \\ &= \frac{2.8}{0.8925 * 2.8094} = \mathbf{1.117} \rightarrow \text{opção } comp \text{ é } \mathbf{1,117} \text{ vezes mais rápida que a opção } base \end{aligned}$$

Relativamente à opção *hard* (f_cpu_hard=2.0Ghz):

$$\begin{aligned} \text{ganho}_{comp \text{ sobre hard}} &= \frac{Texe_{hard}}{Texe_{comp}} = \frac{\#I_{hard} * CPI_{hard} * Tcpu_{hard}}{\#I_{comp} * CPI_{comp} * Tcpu_{comp}} = \\ &= \frac{\#I_{hard} * CPI_{hard} * fcpu_{comp}}{0.8925 * \#I_{hard} * CPI_{comp} * fcpu_{hard}} = \frac{2.45 * 1.5}{0.8925 * 2.8094 * 2} = \mathbf{0,7328} \end{aligned}$$

→ a opção *comp* é mais lenta que a opção *hard*

e) opção *comb* = combinação das opções *hard* e *comp*. CPIcomb = ? e ganho_comb sobre base = ?

CPI's_comb = CPI's_hard ≡ coluna 3 da tabela 4

Frequência das instruções = Freq_comp → #I_comb = #I_comp = 0.8925 * #I_base

Fcpu_comb = Fcpu_hard = 2 GHz

$$CPI_{comb} = \sum_{n=1}^4 Freq_{comp}[n] * cpi_{hard}[n] = 0.4033 * 2 + 0.2521 * 2 + 0.2381 * 3 + 0.1064 * 4 = 2.4509$$

$$\begin{aligned} \text{ganho}_{comb \text{ sobre base}} &= \frac{Texe_{base}}{Texe_{comb}} = \frac{\#I_{base} * CPI_{base} * Tcpu_{base}}{\#I_{comb} * CPI_{comb} * Tcpu_{comb}} = \\ &= \frac{\#I_{base} * CPI_{base} * fcpu_{hard}}{0.8925 * \#I_{base} * CPI_{comb} * fcpu_{base}} = \frac{2.8 * 2}{0.8925 * 2.4509 * 1.5} = \mathbf{1.71} \end{aligned}$$

→ opção *comb* é 1.71 vezes mais rápida que a *base*

f) O tempo para desenvolver cada opção é dado na tabela 6. A concorrência melhora 1.034/mês. Desempenho inicial base = desempenho inicial concorrência. Que opções vale a pena desenvolver?

As opções *hard* e *comp* precisam de 6 meses para estar prontas.

Nessa altura a concorrência será (1.034)⁶ vezes mais rápida, ou seja, **1.222** vezes mais rápida.

→ O ganho da opção *hard* sobre a *base* é **1.524** → **compensa** desenvolve-la.

→ O ganho da opção *comp* sobre a *base* é **1.117**, valor menor que 1.222 → **não** compensa desenvolve-la.

A opção *comb* precisa de 8 meses:

Nessa altura a concorrência será em média (1.034)⁸ vezes mais rápida ≡ **1.3066**

→ Como o ganho de *comb* sobre a *base* é **1.71** → **compensa** desenvolve-la.

Questão 3

a) $CPI_{FP} = ?$

$$CPI_{FP} = 6 * 0.1 + 20 * 0.05 + 4 * 0.3 + 2 * 0.55 = 3.9$$

b) Reduzindo para metade o nº de divisões FP. Qual o ganho de desempenho?

$$CPI_{FP} = 6 * 0.1 + 10 * 0.05 + 4 * 0.3 + 2 * 0.55 = 3.4$$

$$ganho_{b) sobre a)} = CPI_a / CPI_b = 3.9 / 3.4 = 1.147$$

c) Reduzindo para metade o CPI das adições FP. Qual o **ganho**? Explicar o resultado com a lei *amdahl*.

$$CPI = 6 * 0.1 + 20 * 0.05 + 2 * 0.3 + 2 * 0.55 = 3.3$$

$$ganho_{c) sobre a)} = CPI_a / CPI_c = 3.9 / 3.3 = 1.182$$

O ganho é maior porque as adições FP são mais comuns → A lei de *amdahl* diz: "Otimizar o caso mais comum"

d) Suponha outra máquina (NFP) sem unidade vírgula flutuante. Cálculos todos feitos com unidade de inteiros. $MULT_{FP}$ exige 30 ops inteiras. ADD_{FP} exige 20 ops inteiras. DIV_{FP} exige 50 ops inteiras. Quantas vezes é mais rápida a máquina FP em relação a NFP?

O CPI da máquina NFP é 2 → $CPI_{NFP}=2$ (apenas executa operações inteiras).

Mas o número de instruções aumenta:

$$\#I_{NFP} = (0.1 * 30 + 0.05 * 50 + 0.3 * 20 + 0.55 * 1) * \#I_{FP} = 12.05 * \#I_{FP}$$

$$ganho_{FP sobre NFP} = (\#I_{NFP} * CPI_{NFP}) / (\#I_{FP} * CPI_{FP}) = (12.05 * \#I_{FP} * CPI_{NFP}) / (\#I_{FP} * CPI_{FP}) \\ = (12.05 * 2) / 3.9 = 6.179$$

e) Com $f_{cpu_FP} = f_{cpu_NFP} = 1\text{GHz}$, $MIPS_{FP}=?$ e $MIPS_{NFP}=?$

$$MIPS = \#I / (T_{exe} * 10^6) = \#I / (\#I * CPI * T_{cpu} * 10^6) = f_{cpu} / (CPI * 10^6)$$

$$MIPS_{FP} = 10^9 / (3.9 * 10^6) = 10^3 / 3.9 = 256.4$$

$$MIPS_{NFP} = 10^9 / (2 * 10^6) = 10^3 / 2 = 500$$

Conclusão:

O MIPS da máquina NFP é maior que o da FP (quase o dobro), mas o tempo de execução de FP é 6.179 vezes menor.

Isto deve-se ao facto de os programas de NFP terem muitas mais instruções para realizar o mesmo trabalho.