



1. Introdução

Pretende-se que o aluno perceba quais as grandezas envolvidas na avaliação do desempenho de um processador, de que forma estas se inter-relacionam e como é que factores relacionados com a micro-arquitectura do processador e organização da memória as podem influenciar.

2. Avaliação do desempenho sem hierarquia da memória

1. Considere um processador com 3 classes de instruções com o CPI indicado na tabela 1. Um programador tem que seleccionar o compilador a usar para uma determinada aplicação. O número de instruções de cada classe gerado por cada um dos compiladores é apresentado na tabela 2.

Tipo de instrução	CPI
A	1
B	2
C	3

Tabela 1 – CPI por classe de instrução

Compilador	Nº instruções por classe		
	A	B	C
C1	$1 \cdot 10^6$	$3 \cdot 10^6$	$4 \cdot 10^6$
C2	$5 \cdot 10^6$	$2 \cdot 10^6$	$3 \cdot 10^6$

Tabela 2 – Nº de instruções por classe e compilador

- 1.1. Calcule, para cada um dos compiladores o nº de instruções executadas, o CPI global e o número de *clock cycles* necessário para executar a aplicação. Qual o compilador que produz código mais rápido? Quantas vezes mais rápido que o outro compilador?

$$\#I_{C1} = (1+3+4) \cdot 10^6 = 8 \cdot 10^6 \text{ instruções}$$

$$\#I_{C2} = (5+2+3) \cdot 10^6 = 10^7 \text{ instruções}$$

$$CPI_{C1} = (1 \cdot 1 + 3 \cdot 2 + 4 \cdot 3) \cdot 10^6 / 8 \cdot 10^6 = 2.375$$

$$CPI_{C2} = (5 \cdot 1 + 2 \cdot 2 + 3 \cdot 3) \cdot 10^6 / 10^7 = 1.8$$

$$cc_{C1} = \#I_{C1} \cdot CPI_{C1} = 8 \cdot 10^6 \cdot 2.375 = 19 \cdot 10^6$$

$$cc_{C2} = \#I_{C2} \cdot CPI_{C2} = 10^7 \cdot 1.8 = 18 \cdot 10^6$$

C2 produz código $19/18 = 1.056$ vezes mais rápido que C1.

- 1.2. Se a frequência de relógio deste processador é de 500 MHz, qual o tempo de execução da aplicação?

$$\text{Texec}_{C1} = \text{clockcycles}_{C1} / f_{\text{CPU}} = 19 \cdot 10^6 / 500 \cdot 10^6 = 0.038 \text{ s}$$

$$\text{Texec}_{C2} = \text{clockcycles}_{C2} / f_{\text{CPU}} = 18 \cdot 10^6 / 500 \cdot 10^6 = 0.036 \text{ s}$$

- 1.3. Com algumas alterações da organização do processador e da tecnologia usada para o construir, uma equipa de projectistas conseguiu aumentar a frequência do relógio para 1 GHz. No entanto, o CPI de cada uma das classes de instruções aumentou, conforme ilustrado na tabela 3. Calcule o tempo de execução de cada uma das seqüências de código.

Tipo de instrução	CPI
A	2
B	3
C	4

Tabela 3 – CPI por classe de instrução

$$\text{CPI}_{C1} = (1 \cdot 2 + 3 \cdot 3 + 4 \cdot 4) \cdot 10^6 / 8 \cdot 10^6 = 3.375 \quad \text{CPI}_{C2} = (5 \cdot 2 + 2 \cdot 3 + 3 \cdot 4) \cdot 10^6 / 10^7 = 2.8$$

$$\text{cc}_{C1} = \#I_{C1} \cdot \text{CPI}_{C1} = 8 \cdot 10^6 \cdot 3.375 = 27 \cdot 10^6 \quad \text{cc}_{C2} = \#I_{C2} \cdot \text{CPI}_{C2} = 10^7 \cdot 2.8 = 28 \cdot 10^6$$

$$\text{Texec}_{C1} = \text{cc}_{C1} / f_{\text{CPU}} = 27 \cdot 10^6 / 10^9 = 0.027 \text{ s} \quad \text{Texec}_{C2} = \text{cc}_{C2} / f_{\text{CPU}} = 28 \cdot 10^6 / 10^9 = 0.028 \text{ s}$$

Nestas circunstâncias o código produzido por C1 é $28/27 = 1.037$ vezes mais rápido do que aquele produzido por C2, conseguindo-se uma melhoria de $36/27 = 1.33$ vezes relativamente à melhor solução anterior.

- 1.4. Lembrando que o MIPS nativo é dado por $\text{MIPS} = \#I / (\text{Texec} \cdot 10^6)$, qual o MIPS nativo obtido por cada compilador para esta máquina? E o MIPS de pico (*peak*) da máquina?

$$\text{MIPS}_{C1} = 8 / 0.027 = 296,3$$

$$\text{MIPS}_{C2} = 10 / 0.028 = 357,14$$

Para calcular o MIPS de pico note-se que este é obtido com as instruções com mais baixo CPI (2, neste caso) e

$$\text{MIPS} = \frac{\#I}{T_{\text{exec}} \cdot 10^6} \quad ; \quad T_{\text{exec}} = \#I \cdot \text{CPI} \cdot \text{ccT} \quad \text{donde} \quad \text{MIPS} = \frac{10^{-6}}{\text{CPI} \cdot \text{ccT}} = \frac{f \cdot 10^{-6}}{\text{CPI}}$$

$$\text{MIPS}_{\text{pico}} = 10^3 / 2 = 500$$

3. Avaliação do desempenho com hierarquia da memória

2. Considere uma máquina M1, baseada no MIPS, a correr o programa gcc (ver tabela 4), com uma frequência do relógio de 1,333 GHz, com uma *miss rate* de 4% para instruções e 6% para dados. Esta máquina tem uma *miss penalty* de 30 nano segundos para todos os acessos à memória principal (leituras ou escritas) e um CPI_{CPU} de 2.

Tipo instrução	loads	Stores	Tipo-R	branches	jumps
% ocorrência	22 %	11 %	49 %	16 %	2 %

Tabela 4 - percentagem de ocorrência de instruções no gcc para o MIPS

2.1. Quantas vezes mais rápida é M1 do que M2 (a correr o gcc), caracterizada como sendo uma máquina sem *cache* (*miss rate* de instruções e dados=100%)?

Comecemos por calcular a *miss penalty* em ciclos: $mp = 30 * 1,333 = 40$ ciclos

$$\begin{aligned}
 \text{ganho} &= \frac{\text{Texec}_{M2}}{\text{Texec}_{M1}} = \frac{\#I * CPI_{M2} * Tcc}{\#I * CPI_{M1} * Tcc} = \frac{(mrI_{M2} + \%Mem * mrD_{M2}) * mp_{M2} + CPI_{CPUM2}}{(mrI_{M1} + \%Mem * mrD_{M1}) * mp_{M1} + CPI_{CPUM1}} = \\
 &= \frac{(1 + 0,33 * 1) * 40 + 2}{(0,04 + 0,33 * 0,06) * 40 + 2} = \frac{1,33 * 40 + 2}{0,06 * 40 + 2} = \frac{53,2 + 2}{2,4 + 2} = 12,55
 \end{aligned}$$

2.2. Quantas vezes mais rápida é M3 do que M1 (a correr o gcc), sendo M3 uma máquina ideal com uma *cache* infinita (*miss rate* de instruções e dados=0%)?

$$\text{ganho} = \frac{\text{Texec}_{M1}}{\text{Texec}_{M3}} = \frac{0,06 * 40 + 2}{2} = \frac{4,4}{2} = 2,2$$

2.3. Quantas vezes mais rápida é M4 do que M1 (a correr o gcc), sendo M4 uma máquina em tudo idêntica a M1 excepto na frequência do relógio que é de 2 GHz?

Mudando o Tcc (0,75 ns para M1 e 0,5 ns para M4), também muda a *miss penalty* expressa em ciclos para M4: $mp_{M4} = 30 * 2 = 60$ ciclos

$$\text{ganho} = \frac{\text{Texec}_{M1}}{\text{Texec}_{M4}} = \frac{\#I * CPI_{M1} * Tcc_{M1}}{\#I * CPI_{M4} * Tcc_{M4}} = \frac{(0,06 * 40 + 2) * 0,75 * 10^{-9}}{(0,06 * 60 + 2) * 0,5 * 10^{-9}} = \frac{4,4 * 0,75}{5,6 * 0,5} = \frac{3,3}{2,8} = 1,17$$