

# Análise de desempenho

# Análise do desempenho (1)

Avião	Capacidade (passageiros)	Raio de acção (km)	Velocidade (km/h)	Throughput (passageiros × km/h)
Boeing 777	375	7451	982	368 250
Boeing 747	470	6679	982	461 540
BAC/Sud Concorde	132	6437	2172	286 704
Douglas DC-850	146	14033	875	127 750


## Qual o melhor avião?

- ▶ O que permite transportar mais passageiros?
- ▶ O com maior raio de acção?
- ▶ O mais rápido?
- ▶ O que permite transportar mais passageiros mais rapidamente?

# Análise do desempenho (2)

Como medir o desempenho (*performance*) de um computador?

## Possíveis medidas

- ▶ Tempo de execução de um programa (ou de vários) 
  - ▶ Computador pessoal
- ▶ Número de tarefas realizadas por unidade de tempo (*throughput*)
  - ▶ Servidor

# Tempo de execução (1)

O que é o tempo de execução de um programa?

Comummente, é o **tempo decorrido** entre o início e o fim da execução do programa (*wall clock time*, *response time* ou *elapsed time*)

- ▶ pode ser afectado pela carga da máquina
- ▶ em geral, é o apropriado para programas paralelos

Também pode ser o **tempo de CPU**, que é o tempo que o CPU esteve a trabalhar para o programa (*CPU time*) e que se divide em

- ▶ tempo a executar instruções do programa (*user CPU time*) e
- ▶ tempo a executar instruções do sistema operativo em prol do programa (*system CPU time*)

## Tempo de execução (2)

### Exemplo

#### Computador com pouca carga

```
linux$ time ./queens --count-solutions 13
found 73712 solutions

real    0m4.580s      ← tempo decorrido
user    0m4.446s      ← user CPU time
sys     0m0.009s      ← system CPU time
```

#### Computador com alguma carga

```
linux$ time ./queens --count-solutions 13
found 73712 solutions

real    0m9.308s      ← tempo decorrido
user    0m4.349s      ← user CPU time
sys     0m0.005s      ← system CPU time
```

## Comparação do desempenho (1)

Quanto **menor** for o **tempo de execução** de um programa no computador X, **maior** será o **desempenho** de X

$$\text{Desempenho}_X \propto \frac{1}{\text{Tempo de execução}_X}$$

X tem **maior (ou melhor) desempenho** que Y se

$$\text{Desempenho}_X > \text{Desempenho}_Y$$

o que é equivalente a

$$\frac{1}{\text{Tempo de execução}_X} > \frac{1}{\text{Tempo de execução}_Y}$$

ou seja, se

$$\text{Tempo de execução}_X < \text{Tempo de execução}_Y$$

## Comparação do desempenho (2)

### Comparação quantitativa

$$\frac{\text{Desempenho}_X}{\text{Desempenho}_Y} = n$$

$n$  é a *melhoria de desempenho (speedup)* que se obtém usando o computador X, em relação a usar o Y

$$\text{speedup}_{X \leftarrow Y} = \frac{\text{Desempenho}_X}{\text{Desempenho}_Y} = \frac{\text{Tempo de execução}_Y}{\text{Tempo de execução}_X} = n$$

Se  $n > 1$ , o computador X é  *$n$  vezes mais rápido* que o Y

Se  $n < 1$ , tem-se um *slowdown*

## Comparação do desempenho (3)

### Exemplo

Se o computador C corre um programa em 10 s ( $t_C$ ) e o D corre o mesmo programa em 15 s ( $t_D$ ), quantas vezes é C mais rápido que D?

$$speedup_{C \leftarrow D} = \frac{\text{Desempenho}_C}{\text{Desempenho}_D} = \frac{t_D}{t_C} = \frac{15 \text{ s}}{10 \text{ s}} = 1.5$$

C é 1.5 vezes mais rápido que D **para este programa!**



# Relógio

O funcionamento do processador é regulado através de um (sinal de) relógio

Um relógio caracteriza-se pelo seu período, que é o tempo de duração de um ciclo

## Exemplo

$$T = 250 \text{ ps (picossegundos)} = 250 \times 10^{-12} \text{ s} = 0.25 \times 10^{-9} \text{ s}$$

A frequência do relógio (*clock rate*) é o número de ciclos do relógio por segundo, logo é o inverso do período

## Exemplo

$$f = \frac{1}{T} = \frac{1}{0.25 \times 10^{-9} \text{ s}} = 4 \times 10^9 \text{ s}^{-1} = 4 \times 10^9 \text{ Hz} = 4 \text{ GHz}$$

## Desempenho do CPU (1)

Tempo de CPU ( $t_{\text{CPU}}$ ) = N° de ciclos  $\times$  Duração de 1 ciclo ( $T$ )

ou

$$\text{Tempo de CPU } (t_{\text{CPU}}) = \frac{\text{N° de ciclos}}{\text{Frequência do relógio } (f)}$$

### Exemplo

Um programa corre em 10s no computador A, cujo relógio tem uma frequência de 2 GHz.

Pretende-se construir um computador B, com um relógio de frequência bastante superior. No entanto, as alterações que isso implica levam a que o número de ciclos necessários para o programa aumentem em 20%.

Qual terá de ser a frequência do relógio de B para que o programa corra em 6s?

## Desempenho do CPU (2)

### Exemplo (cont.)

Se

$$t_{\text{CPU}_A} = \frac{\text{ciclos}_A}{f_A} \quad \text{e} \quad t_{\text{CPU}_B} = \frac{\text{ciclos}_B}{f_B}$$

então

$$\begin{aligned} \text{ciclos}_A &= t_{\text{CPU}_A} \times f_A \\ &= 10 \times 2 \times 10^9 \\ &= 20 \times 10^9 \text{ ciclos} \quad (20 \text{ mil milhões de ciclos}) \end{aligned}$$

e

$$\begin{aligned} f_B &= \frac{\text{ciclos}_B}{t_{\text{CPU}_B}} = \frac{1.2 \times \text{ciclos}_A}{t_{\text{CPU}_B}} \quad (\text{ciclos}_B = 1.2 \times \text{ciclos}_A) \\ &= \frac{1.2 \times 20 \times 10^9}{6} \\ &= 4 \times 10^9 \text{ Hz} = 4 \text{ GHz} \end{aligned}$$

# Arquitectura do conjunto de instruções

Arquitectura do conjunto de instruções (*instruction set architecture, ISA*) de um processador

- ▶ Geralmente referida somente como **arquitectura**
- ▶ É a especificação das **instruções** que o processador executa e do seu **comportamento**

Um processador **implementa** uma **arquitectura**

# Instruções e desempenho (1)

É possível estimar o **número de instruções** executadas num programa

Sabendo o **número de instruções** executadas e o **tempo médio** que cada uma demora, é possível estimar o tempo que a execução do programa demora

**CPI** (*clock cycles per instruction*) é o tempo médio que uma instrução demora, expresso em ciclos de relógio, *i.e.*, é o **número médio de ciclos por instrução**

$$\text{Nº de ciclos} = \text{Nº de instruções} \times \text{CPI}$$

O CPI pode ser utilizado para comparar diferentes implementações da **mesma** arquitectura

## Instruções e desempenho (2)

### Exemplo

A e B são duas implementações da mesma arquitectura. Em A, o período do relógio é de 250 ps e um programa tem um CPI de 2.0. Em B, estes valores são 500 ps e 1.2, respectivamente. Qual é o computador mais rápido e por quanto?

Se  $N_I$  for o número de instruções executadas no programa

$$\text{ciclos}_A = \text{instruções}_A \times \text{CPI}_A = N_I \times 2.0$$

$$\text{ciclos}_B = \text{instruções}_B \times \text{CPI}_B = N_I \times 1.2$$

O tempo de CPU em cada computador será

$$\begin{aligned} t_{\text{CPU}_A} &= \text{ciclos}_A \times T_A \\ &= N_I \times 2.0 \times 250 \text{ ps} = 500 \times N_I \text{ ps} \end{aligned}$$

$$t_{\text{CPU}_B} = N_I \times 1.2 \times 500 \text{ ps} = 600 \times N_I \text{ ps}$$

Qual o computador mais rápido? E por quanto?

## Instruções e desempenho (3)

### Exemplo (cont.)

$$\frac{\text{Desempenho do CPU}_A}{\text{Desempenho do CPU}_B} = \frac{t_{\text{CPU}_B}}{t_{\text{CPU}_A}} = \frac{600 \times N_I \text{ ps}}{500 \times N_I \text{ ps}} = 1.2$$

A é 1.2 vezes mais rápido que B *para este programa*

Quando se passa de B para A, obtém-se um *speedup* de 1.2

## CPI de um conjunto de instruções

Uma arquitectura inclui várias **classes** de instruções

- ▶ Aritméticas
- ▶ Acesso à memória
- ▶ Saltos (condicionais ou não)
- ▶ ...

As diferentes classes podem apresentar CPIs diferentes

### Exemplo

As instruções de um programa apresentam a seguinte distribuição:

Classe	Aritméticas	Memória	Saltos
CPI	1	3	2
%	60	30	10

$$\begin{aligned}\text{CPI global} &= \sum_{\forall \text{ Classe } C} \%_C \times \text{CPI}_C \\ &= 60\% \times 1 + 30\% \times 3 + 10\% \times 2 \\ &= 1.7\end{aligned}$$



# Equação clássica do desempenho do CPU

Tempo de CPU = N° de instruções  $\times$  CPI  $\times$  Duração de 1 ciclo

$$\text{Tempo de CPU} = \frac{\text{N° de instruções} \times \text{CPI}}{\text{Frequência do relógio}}$$

Factores de desempenho	O que é medido
Tempo de CPU: $t$	Segundos a executar o programa
Número de instruções	Instruções executadas para o programa
Ciclos por instrução: CPI	Número médio de ciclos por instrução
Duração de 1 ciclo: $T$	← Segundos por ciclo de relógio
Frequência do relógio: $f$	← Ciclos de relógio por segundo

# Falácias e armadilhas (1)

## A lei de Amdahl

### Exemplo

Se um programa corre em 100 s e a multiplicação é responsável por 80 s, quanto é necessário aumentar a velocidade da multiplicação para que o programa corra 5 vezes mais depressa?

### Lei de Amdahl

Tempo depois da melhoria =

$$\frac{\text{Tempo afectado pela melhoria}}{\text{Valor da melhoria}} + \text{Tempo não afectado pela melhoria}$$

Neste caso

$$\text{Tempo depois da melhoria} = \frac{100 \text{ s}}{5} = \frac{80 \text{ s}}{n} + (100 - 80 \text{ s})$$

# Falácias e armadilhas (2)

A lei de Amdahl

Exemplo (cont.)

Ou seja

$$\frac{100 s}{5} = 20 s = \frac{80 s}{n} + 20 s \quad \equiv \quad 0 = \frac{80 s}{n}$$

A armadilha

*Pensar que a melhoria de um aspecto de um computador levará a um aumento proporcional do desempenho global*

## Falácias e armadilhas (3)

Milhões de instruções por segundo (MIPS)

Considere-se como medida de desempenho de um computador

$$\text{MIPS} = \frac{\text{N}^\circ \text{ de instruções}}{\text{Tempo de execução} \times 10^6}$$

### Exemplo

Para um programa, tem-se

	Computador A	Computador B
Nº de instruções	10 mil milhões	8 mil milhões
Frequência do relógio	4 GHz	4 GHz
CPI	1.0	1.1

- ▶ Qual o computador com maior valor de MIPS?
- ▶ Qual o computador mais rápido?

## Falácias e armadilhas (4)

Milhões de instruções por segundo (MIPS)

Exemplo (cont.)

$$\text{MIPS}_A = \frac{f_A}{\text{CPI}_A \times 10^6} = \frac{4 \times 10^9}{1.0 \times 10^6} = 4000$$

$$\text{MIPS}_B = \frac{f_B}{\text{CPI}_B \times 10^6} = \frac{4 \times 10^9}{1.1 \times 10^6} = 3636$$

O computador A parece mais rápido mas...

$$t_A = \frac{\text{N}^\circ \text{ de instruções}_A}{\text{MIPS}_A \times 10^6} = \frac{10 \times 10^9}{4000 \times 10^6} = 2.5 \text{ s}$$

$$t_B = \frac{\text{N}^\circ \text{ de instruções}_B}{\text{MIPS}_B \times 10^6} = \frac{8 \times 10^9}{3636 \times 10^6} = 2.2 \text{ s}$$

... o computador B executa o programa em menos tempo

# Falácias e armadilhas (5)

Milhões de instruções por segundo (MIPS)

$$\text{MIPS} = \frac{\text{Nº de instruções}}{\text{Tempo de execução} \times 10^6}$$

## Problemas

1. Não é possível usar para comparar computadores com diferentes conjuntos de instruções
2. O valor de MIPS para um computador depende do programa usado para o calcular
3. Se uma nova versão de um programa executa mais instruções e cada instrução é mais rápida, o valor de MIPS pode aumentar mesmo que o desempenho diminua

## A armadilha

*Usar só parte da equação de desempenho para caracterizar o desempenho*