

**MC722**

**Organização de Computadores**  
Teoria e Prática

2012

Prof. Paulo Cesar Centoducatte  
[ducatte@ic.unicamp.br](mailto:ducatte@ic.unicamp.br)  
[www.ic.unicamp.br/~ducatte](http://www.ic.unicamp.br/~ducatte)

MC722  
2.1

**MC722**

**Arquitetura de Computadores**

**Lei de Amdahl, Desempenho**

"DDCA" - (Capítulo 7)  
"COD" - (Capítulo )

MC722  
2.2

**Micro-Arquitetura**

- Lei Amdahl
- Desempenho
  - Tempo de resposta & *Throughput*
  - *Desempenho Relativo*
  - *Medida de Desempenho*
  - *CPI - Ciclos por Instrução (média)*
  - **Componentes Básicos de Desempenho**

MC722  
2.3

**Lei Amdahl**

- Abordagem Quantitativa
  - Faça o caso comum ser mais rápido
  - Lei de Amdahl:
    - » Relaciona o speedup total de um sistema com o speedup de uma porção do sistema

O speedup no desempenho obtido por uma melhoria é limitado pela fração do tempo na qual a melhoria é utilizada

MC722  
2.4

**Lei Amdahl**

- Speedup devido a uma melhoria E:

$$\text{Speedup}(E) = \frac{\text{Execution\_Time\_Without\_Enhancement}}{\text{Execution\_Time\_With\_Enhancement}} = \frac{\text{Performance\_With\_Enhancement}}{\text{Performance\_Without\_Enhancement}}$$

Suponha que a melhoria E acelera a execução de uma fração F da tarefa de um fator S e que o restante da tarefa não é afetado pela melhoria E.

MC722  
2.5

**Lei Amdahl**

$$\text{ExTime}_{\text{new}} = \text{ExTime}_{\text{old}} \times \left[ (1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right]$$

$$\text{Speedup}_{\text{overall}} = \frac{\text{ExTime}_{\text{old}}}{\text{ExTime}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

MC722  
2.6

### Lei Amdahl

- Exemplo: Suponha que as instruções de ponto flutuante foram melhoradas e executam 2 vezes mais rápidas, porém somente 10% das instruções, em um programa, são FP

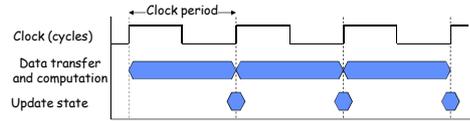
$$\text{ExTime}_{\text{new}} = \text{ExTime}_{\text{old}} \times (0.9 + 0.1/2) = 0.95 \times \text{ExTime}_{\text{old}}$$

$$\text{Speedup}_{\text{overall}} = \frac{1}{0.95} = 1.053$$

MCT22 2.7

### CPU Clocking

- Operation of digital hardware governed by a constant-rate clock



MCT22 2.8

### Lei Amdahl

- Para N Processadores

Fraction<sub>enhanced</sub> = parallelizable part of program

Speedup<sub>enhanced</sub> = n

$$\text{ExTime}_{\text{new}} = \text{ExTime}_{\text{old}} (1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{ExTime}_{\text{old}} \times \text{Fraction}_{\text{enhanced}}}{n}$$

$$\text{Speedup}_{\text{overall}} = \frac{\text{ExTime}_{\text{old}}}{\text{ExTime}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

$$\lim_{n \rightarrow \infty} \text{Speedup}_{\text{overall}} = 1 / (1 - \text{Fraction}_{\text{enhanced}})$$

MCT22 2.9

### Desempenho

- Introdução

- Como medir o desempenho ?

AVIÃO	PASSAGEIROS	AUTOMIA (milhas)	VELOCIDADE (mph)	THROUGHPUT (passag. X veloc)
Boeing 777	375	4630	610	228.750
Boeing 747	470	4150	610	286.700
Concorde	132	4000	1350	178.200
Douglas DC-8	146	8720	544	79.424

O desempenho pode ser definido de diferentes formas, p. ex., velocidade, n. de passageiros, (n. passag. X veloc.) etc.

MCT22 2.10

### Desempenho

- Em computação:

- Um programa sendo executado em duas *workstations* diferentes, a mais rápida é aquela que acaba o *job* primeiro.

» para o usuário → tempo de resposta ou tempo de processamento.

- Em um centro de computação com 2 computadores grandes com *timeshared*, executando *jobs* de vários usuários, o mais rápido é aquele que executa mais *jobs* durante um intervalo de tempo.

» para o adm. de sistemas → throughput → job/hora.

MCT22 2.11

### Desempenho

- Tempo de resposta & Throughput

- Quais das afirmações abaixo faz com que cresça o *throughput*, decresça o tempo de resposta, ou ambos?

» 1 - Mudar o processador por um mais rápido.

» 2 - Adicionar mais um processador ao sistema que usa múltiplos processadores (um para cada tarefa).

(Quase sempre que se decresce o tempo de resposta, o *throughput* cresce).

MCT22 2.12

## Desempenho

### • Resposta

- Em 1 O tempo de resposta e o *throughput* melhoram, enquanto que em 2, o tempo de resposta continua o "mesmo" e o *throughput* cresce.

- Obs.: Diminuindo o tempo de espera, podemos também melhorar o tempo de resposta.

MCT22  
2.13

## Desempenho

### • Para uma máquina X:

- para uma determinada tarefa:

$$\gg \text{Desempenho}_X = (1 / \text{tempo de execução}_X)$$

### • Comparando 2 máquinas X e Y, se:

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

•  $\text{Tempo de execução}_Y > \text{Tempo de execução}_X$

MCT22  
2.14

## Desempenho

### • Desempenho relativo

-  $(\text{Desempenho}_X / \text{Desempenho}_Y)$

$$\gg D_X / D_Y = (\text{Tempo de execução}_Y / \text{Tempo de execução}_X)$$

- Se X é  $n$  vezes mais rápido que Y, então o tempo de execução em Y é  $n$  vezes maior que em X.

### • Exemplo

- Um programa leva 10 segundos na máquina A e 15 na B:  
-  $n = (\text{Tempo de execução}_B / \text{Tempo de execução}_A) = 1.5$

- A é 1.5 vezes mais rápido que B.

MCT22  
2.15

## Desempenho

### • Medida de Desempenho $\rightarrow$ tempo

- Tempo de Execução  $\rightarrow$  segundos/programa

- Tempo de relógio (*clock time*)
- Tempo de resposta (*response time*)
- Tempo transcorrido (*elapsed time*)
- Tempo de CPU (*CPU time*)

#### • OBS.:

• *elapsed time* = tempo de tudo (CPU + I/O + etc.)

$\gg$  *CPU time* = user CPU time + system CPU time (inicialmente só consideraremos o user CPU time)

• *Clock time*  $\rightarrow$  período do clock (clock cycle)  $\rightarrow$  segundos

Ex.: 2nseg

$\gg$  Frequência do clock (clock rate)  $\rightarrow$  Hz

Ex.: 500 MHz

MCT22  
2.16

## Desempenho

### • Tempo de CPU

$T_{CPU}(p/\text{programa}) = \text{períodos de clock da CPU} \times \text{período do clock}$

$T_{CPU}(p/\text{um programa}) = \text{períodos de clock da CPU} (p/\text{um programa}) / \text{frequência do clock}$

### • Exemplo

- Um programa roda em 10 seg. na máquina A, cuja frequência de seu clock é de 400 MHz. Uma máquina B, a ser projetada, tem que rodar este programa em 6 seg. Que acréscimo na frequência de clock é necessário, sabendo-se que na máquina B haverá um acréscimo de 1.2 vezes no número de períodos de clock em relação à máquina A.

MCT22  
2.17

## Desempenho

### • Solução

- A  $\rightarrow$  10 seg.  $\rightarrow$  400 MHz  $\rightarrow$  k períodos

- B  $\rightarrow$  6 seg.  $\rightarrow$  ?  $\rightarrow$  1.2 k períodos

$$\gg t_A = 10 = k/400 \rightarrow k = 4000$$

$$\gg t_B = 6 = 1.2k/x \rightarrow x = (1.2 \times 4000)/6 = 800 \text{ MHz}$$

• B tem que ter um clock com frequência duas vezes maior que A.

MCT22  
2.18

## Desempenho

- nº de períodos da CPU ( para um programa)
  - nº de instruções X nº médio de períodos por instrução (CPI)
  - $T_{CPU}(\text{para um programa}) = (\text{nº de instruções} \times \text{CPI}) / f_{ck}$
  - $T_{CPU}(\text{para um programa}) = (\text{nº de instruções} \times \text{CPI}) \times t_{ck}$

### Exemplo

- Duas implementações para o mesmo Instruction Set.
- Para um determinado programa temos:
  - » Máquina A  $t_{ck}$  CPI
  - » Máquina A 1 ns 2.0
  - » Máquina B 2 ns 1.2

• Qual a máquina mais rápida ? E quanto ?

MCT22 2.19

## Desempenho

Solução:

$I = \text{n. de instruções do programa}$

$$T_{cpuA} = \frac{\text{Tempo da CPU}}{\text{Períodos de clock para o programa}} = I \times 2.0 \times 1n = 2.0n \times I \rightarrow + \text{rápida}$$

$$T_{cpuB} = I \times 1.2 \times 2n = 2.4n \times I$$

$$(\text{CPU performance A} / \text{CPU performance B}) = (2.4n \times I / 2.0n \times I) = 1.2 \rightarrow 1.2 \text{ vezes} + \text{rápida}$$

MCT22 2.20

## Desempenho - Componentes Básicos

COMPONENTES	UNIDADE DE MEDIDA
tempo de CPU	segundos/programa
n. de instruções	instruções executadas (do programa)
períodos de clock/instruções (CPI)	média do n. de períodos de clock para a execução das instruções
período do clock	segundos

- Para instruções de diferentes tipos

$$\text{Períodos de clock da CPU} = \sum_{i=1}^n (CPI_i \times C_i)$$

$n$  → n. de classes de instruções  
 $C_i$  → n. de instruções da classe i  
 $CPI_i$  → média do n. de períodos para instruções da classe i

MCT22 2.21

## Desempenho

### Exemplo:

- Um projetista de compiladores está tentando decidir entre 2 seqüências de códigos para uma determinada máquina. Os aspectos de hardware da máquina são:

classes de instruções	CPI para a classe
A	1
B	2
C	3

- Para um comando, o projetista está considerando 2 seqüências de código:

seqüência de código	n. de instruções para cada classe		
	A	B	C
1	2	1	2
2	4	1	1

Qual a seqüência que executa mais instruções ?

Qual a seqüência mais rápida ? Qual a CPI de cada seqüência ?

MCT22 2.22

## Desempenho

Solução

A seqüência 1 executa :  $2 + 1 + 2 = 5$  instruções → *menos instruções*

A seqüência 2 executa :  $4 + 1 + 1 = 6$  instruções

$$CPU_{\text{períodos de clock}} = CPU_{pc} = \sum_{i=1}^n (CPI_i \times C_i)$$

$$CPU_{pc1} = (2 \times 1) + (1 \times 2) + (2 \times 3) = 10 \text{ períodos}$$

$$CPU_{pc2} = (4 \times 1) + (1 \times 2) + (1 \times 3) = 9 \text{ períodos} \rightarrow \text{mais rápida}$$

$$CPI = CPU_{pc} / \text{n. de instruções}$$

$$CPI_1 = 10 / 5 = 2 \text{ (média de 2 períodos por instrução)}$$

$$CPI_2 = 9 / 6 = 1.5 \text{ (média de 1.5 períodos por instrução)}$$

MCT22 2.23

## Desempenho

Exemplo de utilização de um conjunto de programas para um benchmark:

	Computador A	Computador B
Programa 1	1	10
Programa 2	1000	100
Total (segs)	1001	110

Individualmente:

- A é 10 vezes mais rápido que B para o programa 1
- B é 10 vezes mais rápido que A para o programa 2

Qual o mais rápido ?

Usando o tempo de execução total :

MCT22 2.24

## Desempenho

Usando o tempo de execução total :

$$\begin{aligned} \text{Performance}_B / \text{Performance}_A &= \\ &= \text{Tempo de execução}_A / \text{Tempo de execução}_B = \\ &= 1001 / 110 = 9.1 \end{aligned}$$

→ B é 9.1 mais rápido que A para os programas 1 e 2 juntos !

MCT22  
2.25