

# Desempenho

# Histórico de revisões

2

Revisão	Data	Responsável	Descrição
0.1	- X -	Prof. Cesar Zeferino	Primeira versão
0.2	03/2016	Prof. Cesar Zeferino	Revisão do modelo e atualização de conteúdo
0.3	03/2017	Prof. Cesar Zeferino	Atualização de conteúdo
0.4	06/2020	Prof. Cesar Zeferino	Atualização de referências

**Observação:** Este material foi produzido por pesquisadores do Laboratório de Sistemas Embarcados e Distribuídos (LEDS – Laboratory of Embedded and Distributed Systems) da Universidade do Vale do Itajaí e é destinado para uso em aulas ministradas por seus pesquisadores.

# Introdução

## ❑ Objetivo

- ❑ Apresentar e aplicar os conceitos básicos e as métricas para avaliação de desempenho de processadores e sistemas computacionais

## ❑ Conteúdo

- ❑ Definições básicas
- ❑ Métricas de desempenho
- ❑ Avaliação de desempenho
- ❑ Benchmarks
- ❑ Lei de Amdahl

# Introdução

4

## □ Bibliografia

- PATTERSON, David A.; HENNESSY, John L. O papel da performance. *In: \_\_\_\_\_*. **Organização e projeto de computadores**: a interface hardware/software. 3. ed. Rio de Janeiro: Campus, 2005. cap. 2.

# Introdução

5

## ❑ Lista de abreviaturas e símbolos

- ❑ HW      Hardware
- ❑ SW      Software
- ❑ IS      Instruction Set (Conjunto de Instruções)
- ❑ SO      Sistema Operacional
- ❑ I/F      Interface
- ❑ CPI      Ciclos por instrução
- ❑ I, C      Contagem de instruções

# Por que avaliar o desempenho?

- ❑ A eficiência de um sistema computacional depende do desempenho do HW
- ❑ É impossível determinar a velocidade de execução de um SW pela análise do conjunto de instruções (IS – Instruction Set)
- ❑ Para diferentes aplicações, precisa-se fazer diferentes medições de desempenho
- ❑ Ao escolher um computador é preciso entender como medir o desempenho e conhecer as limitações dessas medições

Desempenho = *Performance*

# Para que entender o desempenho?

- ❑ Para entender porque um programa funciona de determinada maneira
- ❑ Porque um conjunto de instruções pode ser implementado para funcionar melhor do que outro
- ❑ Para saber como algum recurso (CPU, Memória, E/S) afeta o desempenho do computador

# Definindo o desempenho

8

❑ Qual automóvel tem melhor desempenho?





# Definindo o desempenho

9

❑ Qual automóvel tem melhor desempenho?



# Definindo o desempenho

10

❑ Qual automóvel tem melhor desempenho?



# Definindo o desempenho

11

## ❑ Qual aeronave tem melhor desempenho?

Aeronave	Capacidade (Passageiros)
Boeing 777	375
Boeing 747	470
Concorde	132
Douglas DC-8-50	146



# Definindo o desempenho

12

## ❑ Qual aeronave tem melhor desempenho?

Aeronave	Capacidade (Passageiros)	Autonomia (milhas)	Velocidade (mph)	Vazão de passageiros (Passageiros x mph)
Boeing 777	375	4.630	610	222.750
Boeing 747	470	4150	610	286.700
Concorde	132	4.000	1350	178.200
Douglas DC-8-50	146	8.720	544	79.424

# Definindo o desempenho

13

- ❑ **Desktop x CPD/TI** (tempo de resposta x vazão)
  - ❑ O usuário de um desktop busca reduzir o tempo para executar um programa – **tempo de resposta** (ou tempo de execução)
  - ❑ O gerente de um CPD/TI quer maximizar a quantidade de tarefas executadas durante um dia – **vazão**
- ❑ **Tempo de resposta e vazão** são métricas diferentes
- ❑ O estudo do desempenho de **CPU** é focado no **tempo de resposta** enquanto que o do desempenho de **E/S** é focado na **vazão**

# Desempenho em função do tempo de execução

## ❑ Considerando um computador X

$$\text{Desempenho}_X = \frac{1}{T_{\text{execução}_X}}$$

## ❑ Considerando dois computadores X e Y

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

## ❑ Comparando o desempenho de X e Y

- ❑  $n > 1$        $\text{Desempenho}_X > \text{Desempenho}_Y$
- ❑  $n = 1$        $\text{Desempenho}_X = \text{Desempenho}_Y$
- ❑  $n < 1$        $\text{Desempenho}_X < \text{Desempenho}_Y$

# Desempenho relativo

## ❑ Exemplo – Compare o desempenho de A e B

Dados

$$T_{\text{execução}_A} = 10 \text{ s}$$

$$T_{\text{execução}_B} = 15 \text{ s}$$

logo

$$\frac{\text{Desempenho}_A}{\text{Desempenho}_B} = \frac{T_{\text{execução}_B}}{T_{\text{execução}_A}} = \frac{15}{10} = 1,5$$

ou seja

- ❑ A é 1,5 vezes mais rápido que B
- ❑ B é 1,5 vezes mais lento que A

## ❑ Terminologia

- ❑ Melhora-se o desempenho aumentando-o
- ❑ Melhora-se o tempo de resposta diminuindo-o

# Detalhamento: Tempo Real

16

- ❑ **Em muitas aplicações, o desempenho é caracterizado por limitações de tempo real e certos eventos específicos da aplicação precisam ocorrer dentro de um período limitado**
- ❑ **Existem dois tipos básicos de limitações de tempo real**
  - ❑ **Tempo real crítico – Hard RT (ex. ABS)**
    - ❑ A CPU que controla o ABS precisa responder dentro de um tempo rígido partir do momento que ele recebe um sinal indicando que as rodas estão travadas
  - ❑ **Tempo real não-crítico – Soft RT (ex. DVD player)**
    - ❑ Ao manipular quadros de vídeo em um sistema de reprodução de DVD, é aceitável descartar um quadro se isso ocorrer muito raramente



# Métricas de desempenho

17

## ❑ Tempo de execução

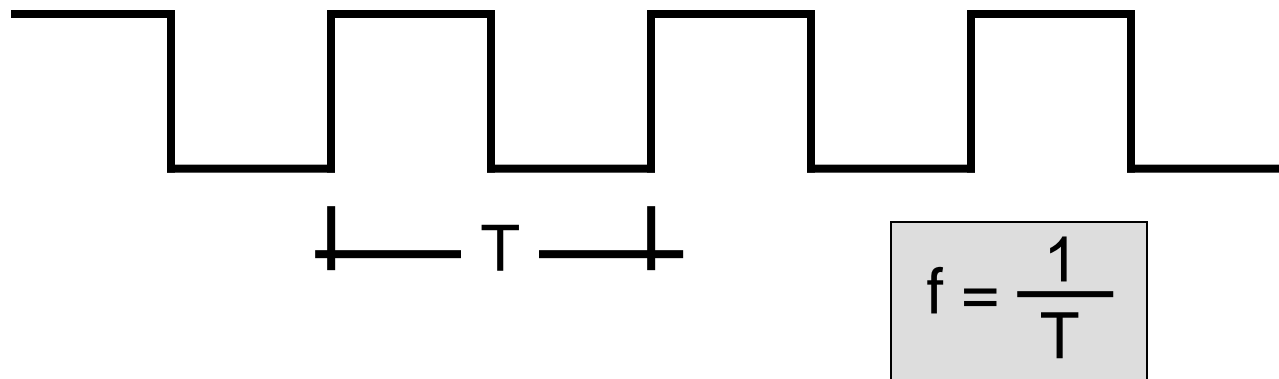
- ❑ Tempo total para completar uma tarefa
- ❑ Inclui tempo de CPU, acesso ao HD, atividade de E/S, overhead do sistema operacional
- ❑ Medido em segundos por programa
- ❑ Também chamado de Tempo de resposta, Tempo de relógio ou Tempo decorrido

# Métricas de desempenho

- ❑ **Tempo de execução da CPU (ou tempo de CPU)**
  - ❑ Tempo que a CPU gasta para uma tarefa específica
  - ❑ Importante em sistemas multitarefa
  - ❑ Pode ser dividido em
    - ❑ Tempo de CPU do usuário
    - ❑ Tempo de CPU do sistema
- ❑ **Importante**
  - ❑ O desempenho do sistema refere-se ao tempo de execução
  - ❑ O desempenho da CPU refere-se ao tempo da CPU

# Revisão: Clock

## ❑ Sinal de clock



## ❑ Terminologia

- ❑  $T$  = ciclo ou período de relógio (medido em segundos)
- ❑  $f$  = frequência ou velocidade do relógio (medida em Hertz)

## ❑ Importante

$$1 \text{ ms} = 10^{-3} \text{ s}$$

$$1 \text{ } \mu\text{s} = 10^{-6} \text{ s}$$

$$1 \text{ ns} = 10^{-9} \text{ s}$$

$$1 \text{ KHz} = 10^3 \text{ Hz}$$

$$1 \text{ MHz} = 10^6 \text{ Hz}$$

$$1 \text{ GHz} = 10^9 \text{ Hz}$$

$$1 \text{ KHz} = 1 / (1 \text{ ms})$$

$$1 \text{ MHz} = 1 / (1 \text{ } \mu\text{s})$$

$$1 \text{ GHz} = 1 / (1 \text{ ns})$$

# Desempenho da CPU e seus fatores

## □ Estimando o tempo de execução da CPU

$$T_{\text{execuçãoda CPU}_{\text{para um programa}}} = \text{Ciclos da CPU}_{\text{para um programa}} \times T_{\text{clock}}$$

ou

$$T_{\text{execuçãoda CPU}_{\text{para um programa}}} = \frac{\text{Ciclos da CPU}_{\text{para um programa}}}{f_{\text{clock}}}$$

## □ Para melhorar o desempenho da CPU

□ Reduzir o número de ciclos para a execução do programa

ou

□ Reduzir o tempo do ciclo de clock (período do clock), ou seja, aumentar a frequência do clock

# Calculando o número de ciclos de clock da CPU

## ❑ Estimando o número de ciclos de clock da CPU

$$\text{Ciclos da CPU}_{\text{para um programa}} = \text{Instruções}_{\text{para um programa}} \times \text{Ciclos por instrução}$$

## ❑ Instruções para um programa (I ou C)

- ❑ É o número de instruções executadas pelo programa

## ❑ Ciclos por instrução (CPI)

- ❑ Número médio de ciclos de relógio gastos pelas instruções
- ❑ Varia de programa para programa
- ❑ Permite comparar duas implementações (organizações) de um mesmo conjunto de instruções (arquitetura)

# Calculando o número de ciclos de clock da CPU (cont.)

22

- ❑ Reescrevendo a equação para estimar o número de ciclos de clock da CPU

$$\text{Ciclos da CPU}_{\text{para um programa}} = I \times \text{CPI}$$

- ❑ Os valores de  $I$  e de  $\text{CPI}$  podem ser determinados por meio de uma simulação detalhada ou pelo uso de contadores de HW na CPU
- ❑ A equação do tempo de execução da CPU pode ser reescrita como

$$T_{\text{execução da CPU}} = I \times \text{CPI} \times T_{\text{clock}}$$

# Usando a equação do desempenho

- ❑ Considerando duas organizações (A e B) para a mesma arquitetura executando um determinado programa com  $I$  instruções, qual é a mais rápida?

Dados:

**CPU<sub>A</sub>**

**CPU<sub>B</sub>**

$$T_{\text{clock}_A} = 250 \text{ ps}$$

$$T_{\text{clock}_B} = 500 \text{ ps}$$

$$\text{CPI}_A = 2,0$$

$$\text{CPI}_B = 1,2$$

**1º) Calculando o número de ciclos de CPU**

$$\text{Ciclos da CPU}_A = I \times 2,0$$

$$\text{Ciclos da CPU}_B = I \times 1,2$$

# Usando a equação do desempenho (continuação)

## 2º) Calculando o tempo de CPU

$$T_{\text{execuçãoda CPU}_A} = \text{Ciclos da CPU}_A \times T_{\text{clock}_A} = 1 \times 2,0 \times 250 \text{ ps} = 500 \times 1 \text{ ps}$$

$$T_{\text{execuçãoda CPU}_B} = \text{Ciclos da CPU}_B \times T_{\text{clock}_B} = 1 \times 1,2 \times 500 \text{ ps} = 600 \times 1 \text{ ps}$$

## 3º) Determinando a relação de desempenho

$$\frac{\text{Desempenho}_{\text{CPU}_A}}{\text{Desempenho}_{\text{CPU}_B}} = \frac{T_{\text{execuçãob}_B}}{T_{\text{execuçãoa}_A}} = \frac{600 \times 1 \text{ ps}}{500 \times 1 \text{ ps}} = 1,2$$

## Conclusão:

- ❑ O computador A é 1,2 vezes (20%) mais rápido do que o computador B



# Ciclos de clock da CPU baseado nas classes de instrução

25

- ❑ Se for possível determinar o CPI e o número de instruções para cada classe de instrução, então o cálculo do número do número de ciclos de clock pode ser mais preciso

$$\text{Ciclos da CPU}_{\text{para um programa}} = \sum_{i=1}^n (I_i \times \text{CPI}_i)$$

onde

- ❑  $I_i$  : Contagem de instruções para instruções da classe  $i$   
(no livro é utilizado  $C_i$ )
- ❑  $\text{CPI}_i$  : CPI para instruções da classe  $i$

# Componentes básicos do desempenho ( $T_{\text{execução da CPU}}$ )

26

Componentes do desempenho	Unidade de medida
Tempo de execução da CPU para um programa	Segundos para o programa
Contagem de instruções (I ou C)	Instruções executadas p/ o programa
Ciclos de clock por instrução (CPI)	Número médio de ciclos por instrução
Tempo do ciclo de clock	Segundos por ciclo de clock

$$T_{\text{execução da CPU}} = I \times \text{CPI} \times T_{\text{clock}}$$

ou seja

$$T_{\text{execução da CPU}} = \frac{\text{Segundos}}{\text{Programa}} = \frac{\text{Instruções}}{\text{Programa}} \times \frac{\text{Ciclos de clock}}{\text{Instrução}} \times \frac{\text{Segundos}}{\text{Ciclo de clock}}$$

# Taxa de execução das instruções

- ❑ Reflete quantas instruções o processador executa em um segundo

$$\text{Taxa}_{\text{execução de instruções}} = \frac{I}{T_{\text{execução da CPU}}}$$

- ❑ Também pode ser obtida pelo CPI e pelo período (ou frequência) do relógio

$$\text{Taxa}_{\text{execução de instruções}} = \frac{1}{\text{CPI} \times T_{\text{clock}}} = \frac{f_{\text{clock}}}{\text{CPI}}$$

As unidades da taxa de execução das instruções são o **IPS** e o **FLOPS**

# Taxa de execução das instruções

28

## ❑ MIPS (Milhões de Instruções por Segundo)

- ❑ Considera o número de instruções executadas, independentemente da classe da instrução
- ❑ Especifica o desempenho de maneira inversa ao tempo de execução (quanto mais MIPS, melhor)

# Taxa de execução das instruções

29

## ❑ MIPS (Milhões de Instruções por Segundo)

### ❑ Limitações

- ❑ Não leva em conta as capacidades das instruções e não se pode comparar computadores com arquiteturas diferentes
- ❑ Varia entre diferentes programas em um mesmo computador; logo, um computador não pode ter um índice MIPS único
- ❑ Pode sofrer com anomalias, variando com o inverso do desempenho - ver Patterson e Hennesy (2005, p. 203-204)

# Taxa de execução das instruções

## ❑ MIPS (Milhões de Instruções por Segundo)

### ❑ Exemplos

Processador	MIPS	Fclk (MHz)
UNIVAC I	0,002	2,25
Zilog Z80	0,580	4
Intel 8086	0,75	10
Microchip PIC16F	5,0	20
ARM Cortex-M3	125,0	100
Intel Core i7 6950X	317.900,0	3000

### ❑ Lista completa em (com linha de tempo)

[http://en.wikipedia.org/wiki/Instructions\\_per\\_second](http://en.wikipedia.org/wiki/Instructions_per_second)

# Taxa de execução das instruções

31

## ❑ FLOPS (Operações de Ponto Flutuante por Segundo)

- ❑ Considera apenas as instruções aritméticas (+, −, × e ÷) que realizam operações sobre números reais representados na notação de ponto flutuante com precisão simples ou dupla (*float, real, double, double precision*)
- ❑ Não leva em conta instruções de transferência e de controle
- ❑ Unidades típicas: MFLOPS e GFLOPS

# Taxa de execução das instruções

32

## ❑ FLOPS (Operações de Ponto Flutuante por Segundo)

### ❑ Limitações

- ❑ Não tem consistência assegurada entre máquinas diferentes, pois algumas, por exemplo, possuem instruções como raiz quadrada e seno em hardware, enquanto outras não e precisam realizar várias operações de ponto flutuante
- ❑ Instruções de ponto flutuante variam em complexidade e em ciclos de execução. Um programa com 100% de somas em PF terá um índice de MFLOPS maior que um programa com 100% de divisões em PF



# Entendendo o desempenho de programas

33

Componentes de SW ou HW	Afeta o quê?	Como?
Algoritmo	I CPI (possivelmente)	O algoritmo determina o número de instruções e pode favorecer instruções com CPI maior
Linguagem de programação	I CPI	<p>A instruções da linguagem são traduzidas para instruções de máquina, afetando a contagem de instruções</p> <p>Linguagens com maior abstração de dados (ex. Java) exigem chamadas indiretas que usam instruções com CPI maior</p>
Compilador	I CPI	Ao efetuar a tradução do programa, o compilador determina as instruções de máquina que serão usadas, afetando o CPI e a contagem de instruções
Arquitetura (conjunto de instruções)	I CPI $T_{\text{clock}}$	A arquitetura determina as instruções disponíveis para execução do programa e, portanto, a contagem de instruções e o CPI. A arquitetura também restringe a implementação, o que impacta na frequência de operação do clock e, portanto, no seu período

# Avaliando o desempenho

34

## ❑ Definições básicas

### ❑ Workload (carga de trabalho)

- ❑ É um conjunto de programas que é a coleção real das aplicações executadas por um usuário (ou é construído a partir dos programas reais para representar essa coleção)

### ❑ Benchmark

- ❑ É uma coleção de programas escolhidos especificamente para realizar a avaliação de desempenho de um computador

# Avaliando o desempenho

35

## ❑ Definições básicas

### ❑ Tipos básicos de benchmark

- ❑ Sintético: Coleção de programas que procuram reproduzir características de aplicações reais
- ❑ Real: Coleção de programas baseada em aplicações reais (ex. compiladores, codificadores,...)

# Avaliando o desempenho

36

## ❑ Quando não usar programas reais?

- ❑ Benchmarks sintéticos pequenos são mais adequados nas fases iniciais do projeto pois
  - ❑ muitas vezes ainda não existem compiladores e a tradução de programas tem que ser manual
  - ❑ podem ser traduzidos e simulados mais facilmente

# Avaliando o desempenho

37

- ❑ **Benchmarks para os diferentes tipos de sistemas**
  - ❑ **Desktop:** foco no desempenho da CPU (ex. SPEC CPU) ou da execução de uma tarefa (ex. jogos)
  - ❑ **Servidores**
    - ❑ **Científicos:** foco no desempenho da CPU
    - ❑ **Outros:** foco no desempenho para Web, compartilhamento de arquivos e banco de dados
  - ❑ **Sistemas embutidos (embarcados):** usa-se aplicações reais ou segmentos delas (ex. EEMBC) - <http://www.eembc.org/home.php>

# Avaliação de desempenho

38

## ❑ Princípio da reprodutibilidade

- ❑ Uma medição de desempenho deve poder ser reproduzida (duplicada) por outra pessoa
- ❑ Os experimentos de medição devem gerar um relatório em que constem informações sobre a plataforma (HW+SW) utilizada
- ❑ O relatório também deve informar quais entradas (dados) foram utilizadas nos experimentos, pois elas tem impacto no fluxo de execução dos programas

# Descrição de um sistema em um relatório de desempenho

39

Hardware	
<b>Fabricante</b>	<b>Dell</b>
<b>Modelo</b>	<b>Precision Wokstation 360</b>
<b>CPU</b>	<b>Intel Pentium 4 (barramento 800 MHz)</b>
<b>CPU MHz</b>	<b>3200</b>
<b>FPU</b>	<b>Integrado</b>
<b>CPU(s) habilitada(s)</b>	<b>1</b>
<b>CPU(s) possível(is)</b>	<b>1</b>
<b>Em paralelo</b>	<b>Não</b>
<b>Cache L1</b>	<b>12KB (I) + 8 KB (D)</b>
<b>Cache L2</b>	<b>512 KB (I+D)</b>
<b>Cache L3</b>	<b>2 MB (I+D)</b>
<b>Memória RAM</b>	<b>4 x 512 MB ECC DDR400 SDRAM CL3</b>
<b>Disco rígido</b>	<b>1 x 90 GB ATA/1000 7200 RPM</b>

# Descrição de um sistema em um relatório de desempenho

40

Software	
Sistema operacional	Windows XP Professional SP1
Compilador	Intel C++ Compiler (20030202Z) Microsoft Visual Studio.NET (7.0.9466) MicroQuill SmartHeap Library 6.01
Sistema de arquivos	NTFS
Estado do sistema	padrão

- ❑ A descrição do sistema pode ainda ser complementada com notas a respeito de flags especiais usados para
  - ❑ portabilidade
  - ❑ otimização
  - ❑ tuning
  - ❑ temporização básica
  - ❑ biblioteca especial
  - ❑ configuração da BIOS
  - ❑ entre outros



# Comparando e resumindo o desempenho

## Exemplo

	Computador A	Computador B
$T_{\text{execução}_{\text{programa1}}}$	1	10
$T_{\text{execução}_{\text{programa2}}}$	1000	100
$T_{\text{execução}_{\text{total}}}$	1001	110

## Afirmativas válidas

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

mas, considerando os dois programas juntos

$$\frac{\text{Desempenho}_B}{\text{Desempenho}_A} = \frac{T_{\text{execução}_A}}{T_{\text{execução}_B}} = \frac{1001}{110} = 9,1$$

- B é 9,1 vezes mais rápido do que A

# Benchmarks SPEC

42

## ❑ SPEC – System Performance Evaluation Corporation

- ❑ Organização fundada (em 1989) e patrocinada por fabricantes de computador (<http://www.spec.org>)
- ❑ Visa criar benchmarks padronizados para avaliar sistemas computacionais, usando aplicações reais e requisitos de relatório rigidamente especificados
- ❑ Inclui benchmarks específicos para avaliação de
  - ❑ CPU
  - ❑ Gráficos
  - ❑ Computação de alto desempenho
  - ❑ Computação orientada a objetos
  - ❑ Aplicações Java
  - ❑ Modelos cliente-servidor
  - ❑ Sistemas de e-mail
  - ❑ Sistemas de arquivo
  - ❑ Servidores web
  - ❑ Computação em nuvem

# Tempo médio de execução

- ❑ Considerando  $n$  programas em um workload pode-se calcular o tempo médio para execução dos programas que o compõem
- ❑ A média aritmética padrão considera que todos são executados com a mesma frequência ( $1/n$ )

$$T_{\text{médio de execução}} = \frac{1}{n} \times \sum_{i=1}^n T_{\text{execução}_{\text{programa } i}}$$

# Tempo médio de execução

44

- A média aritmética ponderada considera que cada programa pode ser executado com uma frequência diferente que representa um percentual específico da carga de trabalho

$$T_{\text{médio de execução}} = \sum_{i=1}^n (w_i \times T_{\text{execução}_{\text{programa } i}})$$

- $w_i$  indica a frequência do programa  $i$  no workload (ex. 0,8 ou 80%)

# Benchmark SPEC CPU2000

- ❑ Incluía 12 programas de inteiros e 14 de ponto flutuante para avaliação de desempenho de CPU, com base no tempo de execução do sistema

Benchmarks de inteiros		Benchmarks de ponto flutuante	
gzip	eon	wupwise	EQUAKE
vpr	perlbnk	swim	facerec
gcc	gap	mgrid	amoo
mcf	vortex	applu	lucas
crafty	bzip2	mesa	fma3d
parser	twolf	galgel	sixtrack
		art	apsi

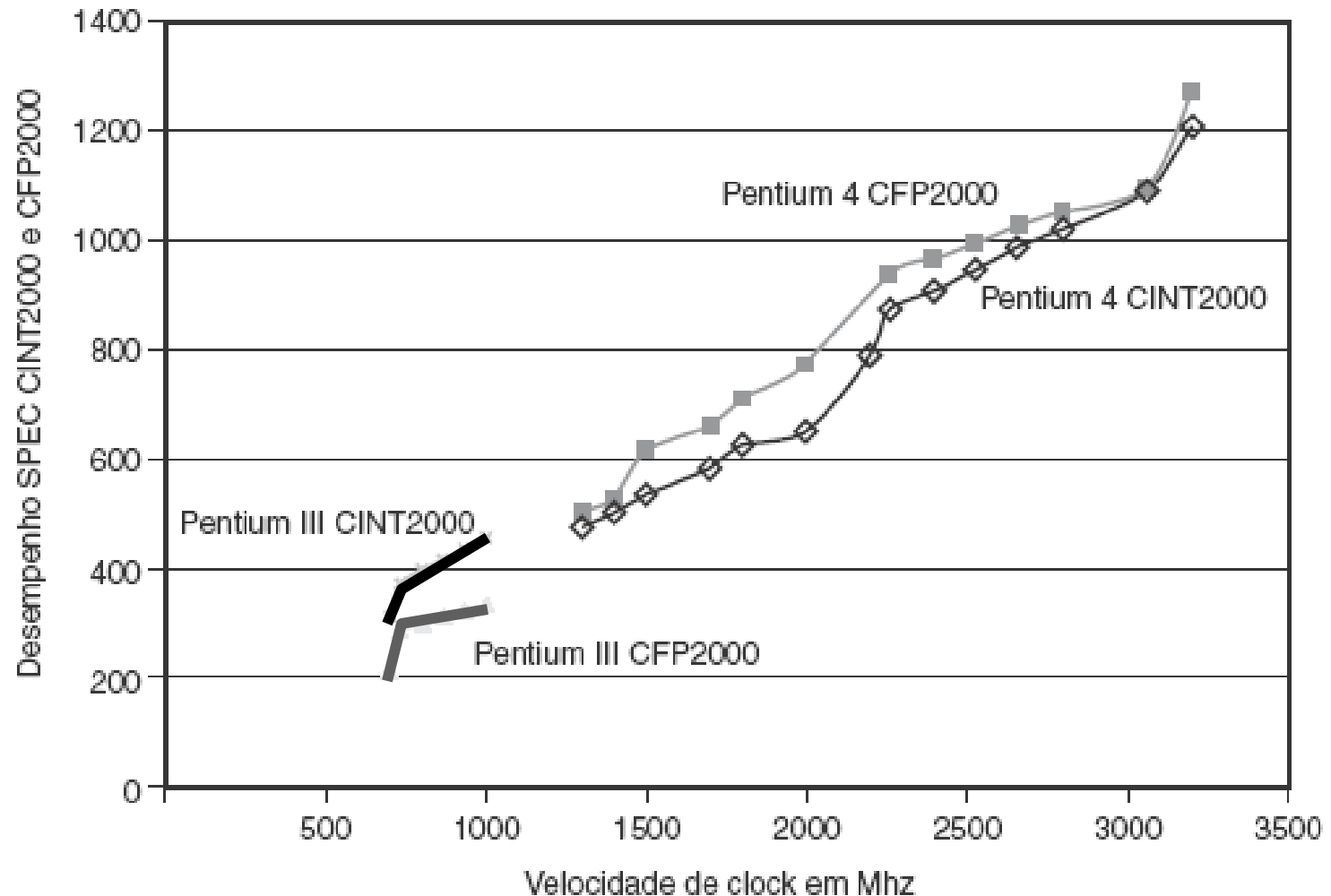
- ❑ Razão SPEC

- ❑ As medições de tempo de execução são normalizadas em relação a uma estação de trabalho Sun Ultra 5\_10 com CPU de 300 MHz
- ❑ CINT2000: média geométrica das razões SPEC de inteiros
- ❑ CFP2000: média geométrica das razões SPEC de ponto flutuante

# Benchmark SPEC CPU2000

46

- ❑ Avaliações SPEC CINT2000 e CFP 2000 para Pentium III e Pentium IV com diferentes clocks



# Benchmark SPEC CPU2006

47

- ❑ **Versão mais atual**
- ❑ **Inclui 12 programas de inteiros e 19 de ponto flutuante**
- ❑ **Métricas**
  - ❑ **SPECspeed**: usada para comparar a capacidade de um computador em executar tarefas únicas
  - ❑ **SPECrate**: mede a vazão (ou taxa) de execução de múltiplas tarefas
- ❑ **Preço (V1.2)**: U\$800,00

# Benchmark SPECweb

48

## ❑ Histórico

- ❑ 1996, primeiro benchmark para medir desempenho de servidores Web
- ❑ 1999, SPECweb99
- ❑ 2005, SPECweb2005
- ❑ 2009, SPECweb2009 (descontinuado em 01/2012)

## ❑ Características do SPECweb99

- ❑ Mede o número máximo de conexões que um sistema funcionando como um servidor Web consegue executar
- ❑ Depende de uma ampla medida de características de sistemas, incluindo o sistema de disco e a rede



# Benchmark SPECweb

49

## ❑ Características do SPECweb2005

- ❑ Mede sessões de usuário simultâneas
- ❑ Implementações PHP e JSP incluídas
- ❑ Inclui múltiplas cargas de trabalho padronizadas atendendo as principais necessidades do mercado
  - ❑ Home Banking (HTTPS), E-commerce (HTTP and HTTPS), e Support (HTTP)
- ❑ Simula o funcionamento das caches dos *browsers*

## ❑ Características do SPECweb2009

- ❑ Versão atualizada do SPECweb2005

# Benchmark SPECweb

50

- Desempenho do SPECweb99 para diversos sistemas Dell PowerEdge usando as versões Xeon (c/ suporte a multiprocessamento) dos processadores Intel



Sistema	CPU	No. de HDs	No. de CPUs	No. de redes	Clock (GHz)	Resultado
1550/1000	Pentium III	2	2	2	1,00	2765
1650	Pentium III	3	2	1	1,40	1910
2500	Pentium III	8	2	4	1,13	3435
2550	Pentium III	1	2	1	1,26	1454
2650	Pentium 4 Xeon	5	2	4	3,06	5698
4600	Pentium 4 Xeon	10	2	4	2,20	4615
6400/700	Pentium III Xeon	5	4	4	0,70	4200
6600	Pentium 4 Xeon MP	8	4	8	2,00	6700
8450/700	Pentium III Xeon	7	8	8	0,70	8001

# Benchmark SPEC Cloud\_IaaS 2016

51

- ❑ 1º benchmark para medir desempenho de infraestrutura de computação em nuvem
- ❑ Foco no desempenho de IaaS (Infrastructure-as-a-Service) de plataformas de computação em nuvem, públicas ou privadas
- ❑ Procura estressar a infraestrutura usando cargas de trabalho intensivas em I/O e CPU
- ❑ Preço (V.1.1): U\$ 2000,00

# Supercomputadores – Top500

52

❑ Site: <https://www.top500.org/>

❑ Lista de 11/2016: <https://www.top500.org/lists/2016/11/>

#	Local	Sistema	Cores	Rmax Tflop/s	Rpeak Tflop/s	kW
1	<a href="#">National Supercomputing Center in Wuxi</a> China	<a href="#">Sunway TaihuLight</a> - <a href="#">Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC</a>	10.649.600	93.014,6	125.435,9	15.371
2	<a href="#">National Super Computer Center in Guangzhou</a> China	<a href="#">Tianhe-2 (MilkyWay-2)</a> - <a href="#">TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P</a> NUDT	3.120.000	33.862,7	54.902,4	17.808
3	<a href="#">DOE/SC/Oak Ridge National Laboratory</a> United States	<a href="#">Titan</a> - <a href="#">Cray XK7</a> , <a href="#">Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x</a>	560.640	17.590,0	27.112,5	8.209

# Supercomputadores – Top500

53

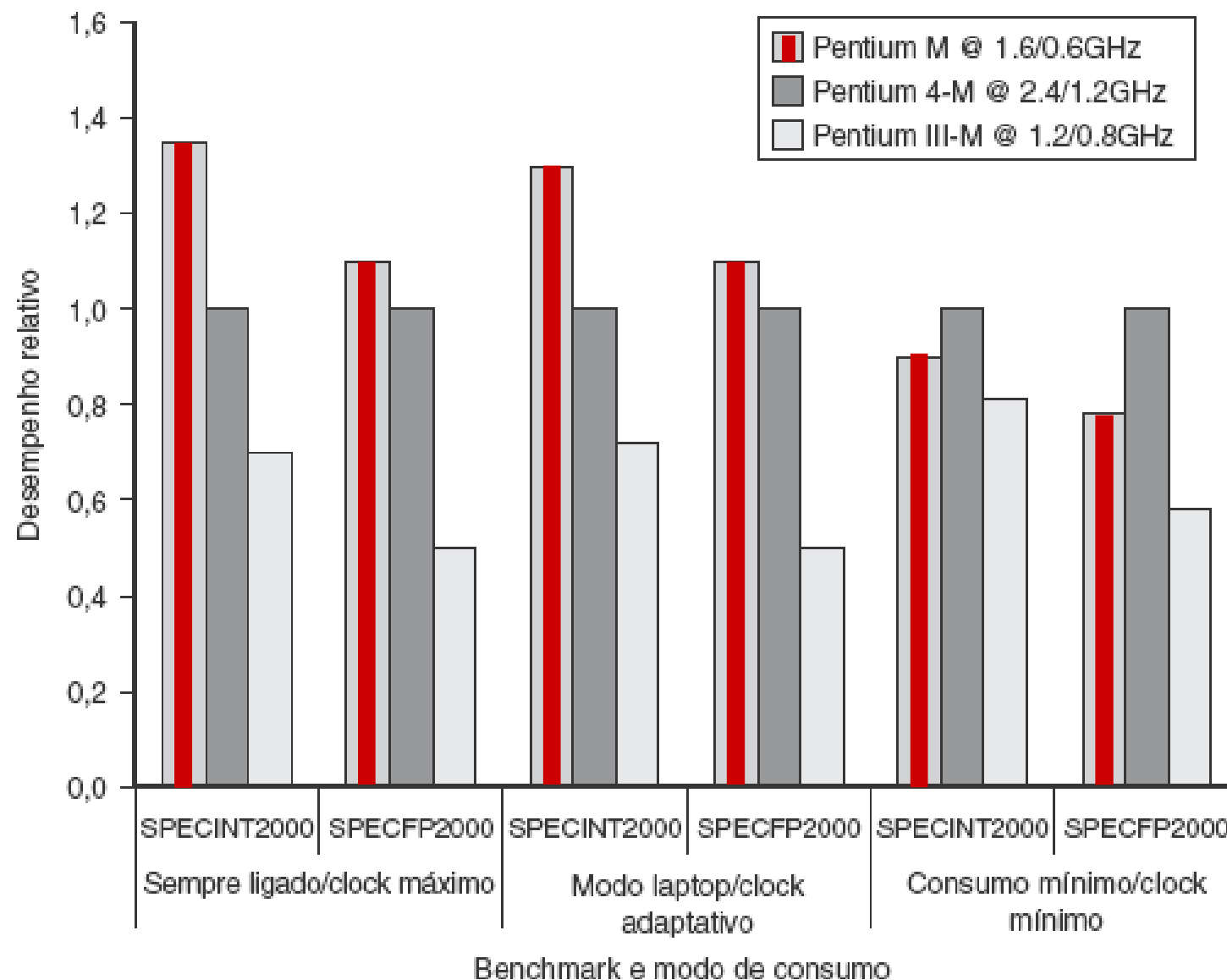
❑ Site: <https://www.top500.org/>

❑ Lista de 11/2016: <https://www.top500.org/lists/2016/11/>

#	Local	Sistema	Cores	Rmax Tflop/s	Rpeak Tflop/s	kW
1	<a href="#">National Supercomputing Center in Wuxi</a> China	<a href="#">Sunway TaihuLight</a> - <a href="#">Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC</a>	10.649.600	93.014,6	125.435,9	15.371
...	...	...	...	..	...	...
364	<a href="#">Laboratório Nacional de Computação Científica</a> Brazil	<a href="#">Santos Dumont GPU</a> - <a href="#">Bullx B710, Intel Xeon E5-2695v2 12C 2.4GHz, Infiniband FDR, Nvidia K40</a> Bull, Atos Group	10.692	456,8	657,8	371

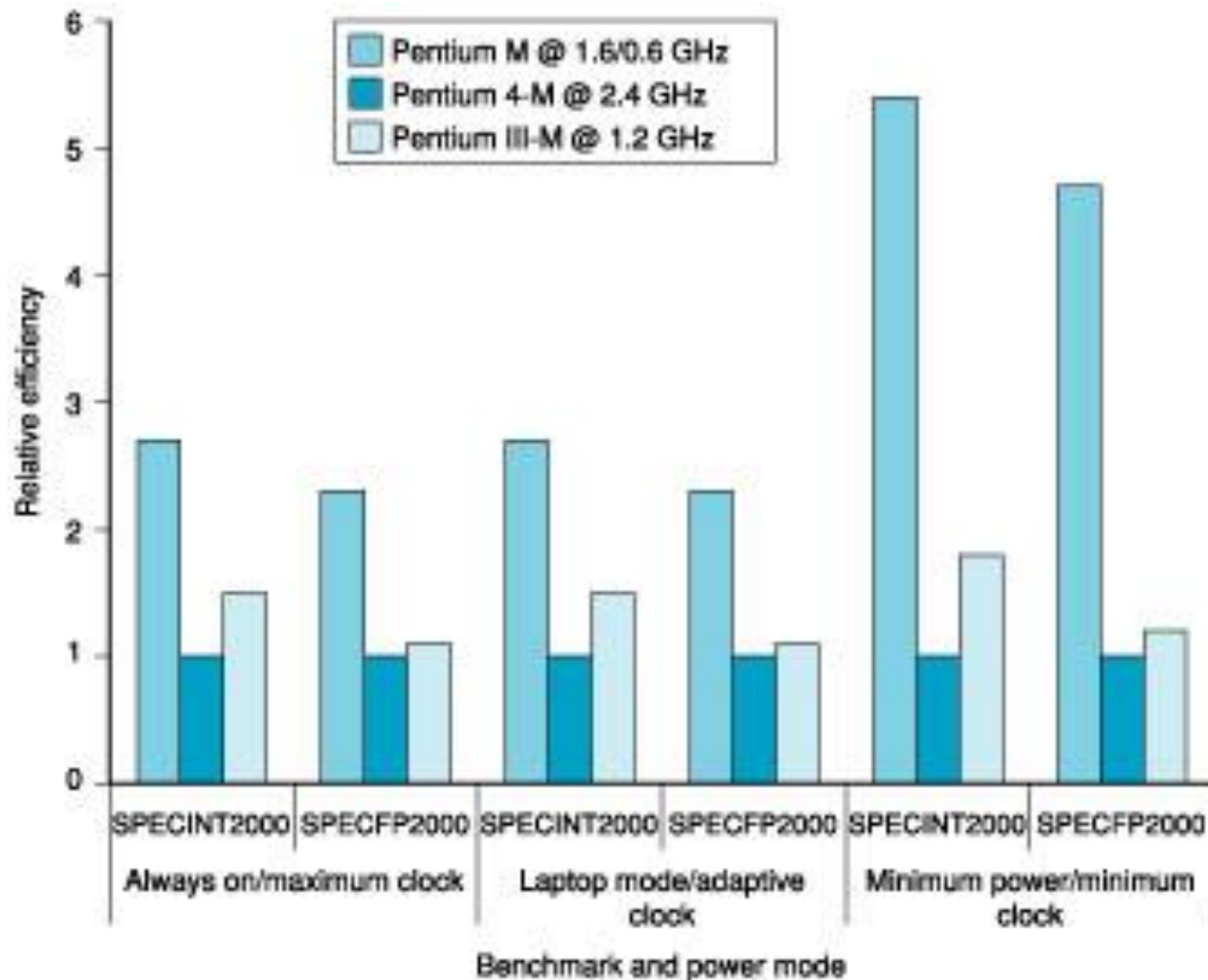
# Desempenho e eficiência do consumo

54



# Eficiência relativa de consumo

55



# Lei de Amdahl

- Permite evidenciar a aceleração (ganho) de desempenho de um computador após a melhoria de alguma de suas características

$$\text{Aceleração} = \frac{T_{\text{execução}_{\text{antigo}}}}{T_{\text{execução}_{\text{novo}}}}$$

$$T_{\text{execução}_{\text{novo}}} = T_{\text{execução}_{\text{antigo}}} \times \left[ \text{Parcela}_{\text{não-usada}} + \frac{\text{Parcela}_{\text{usada}}}{\text{Aceleração}_{\text{usada}}} \right]$$

$$\text{Aceleração} = \frac{T_{\text{execução}_{\text{antigo}}}}{T_{\text{execução}_{\text{novo}}}} = \frac{1}{\left[ \text{Parcela}_{\text{não-usada}} + \frac{\text{Parcela}_{\text{usada}}}{\text{Aceleração}_{\text{usada}}} \right]}$$



# Resumo das equações

## Desempenho

$$\text{Desempenho}_x = \frac{1}{T_{\text{execução}_x}}$$

$$\frac{\text{Desempenho}_x}{\text{Desempenho}_y} = \frac{T_{\text{execução}_y}}{T_{\text{execução}_x}} = n$$

## Tempo de execução da CPU

$$T_{\text{execução da CPU para um programa}} = \text{Ciclos da CPU para um programa} \times T_{\text{clock}}$$

## Taxa de execução de instruções

$$\text{Taxa}_{\text{execução de instruções}} = \frac{1}{\text{CPI} \times T_{\text{clock}}} = \frac{f_{\text{clock}}}{\text{CPI}}$$

# Exercício 1

58

- ☐ Nosso programa favorito é executado em 10 segundos no computador A, que possui um relógio de 4GHz. Estamos tentando ajudar um projetista de computador a construir uma nova máquina B, que execute esse programa em 6 segundos.
- ☐ O projetista determinou que um aumento substancial na velocidade de relógio é possível, mas esse aumento afetará o restante do projeto da CPU, fazendo com que o computador B exija 1,2 vez mais ciclos de clock do que o computador A para esse programa.
- ☐ Que velocidade de relógio devemos pedir para que o projetista almeje?

## Exercício 2

- ❑ **Suponha que tenhamos duas implementações da mesma arquitetura do conjunto de instruções (ISA). Para um determinado programa,**
  - ❑ A máquina A tem um tempo de ciclo de clock de 250 ps e uma CPI de 2,0
  - ❑ A máquina B tem um tempo de ciclo de clock de 500 ps e uma CPI de 1,2
- ❑ **Que máquina é mais rápida para esse programa e o quanto?**
- ❑ **Se duas máquinas possuem a mesma ISA, qual de nossas quantidades (por exemplo, velocidade de relógio, CPI, tempo de execução, número de instruções, MIPS) será sempre idêntica?**

## Exercício 3

- ☐ Um projetista de compilador está tentando decidir entre duas sequências de código para um determinada máquina. Baseado na implementação de hardware, existem três classes diferentes de instruções: Classe A, Classe B e Classe C, e elas exigem um, dois e três ciclos, respectivamente.
- ☐ A primeira sequência de código possui 5 instruções: 2 de A, 1 de B e 2 de C
- ☐ A segunda sequência possui 6 instruções: 4 de A, 1 de B e 1 de C
- ☐ Que sequência será mais rápida? O quanto mais rápida? Qual é a CPI para cada sequência?

# Exercício 4

- ❑ **Dois compiladores diferentes estão sendo testados para uma máquina de 4GHz com três classes diferentes de instruções: Classe A, Classe B e Classe C, e elas exigem um, dois e três ciclos, respectivamente. Ambos os compiladores são usados para produzir código para um grande software.**
- ❑ **O código do primeiro compilador usa 5 milhões de instruções da Classe A, 1 milhão de instruções da Classe B e 1 milhão de instruções da Classe C.**
- ❑ **O código do segundo compilador usa 10 milhões de instruções da Classe A, 1 milhão de instruções da Classe B e 1 milhão de instruções da Classe C.**
- ❑ **Que sequência será mais rápida de acordo com o MIPS? Que sequência será mais rápida de acordo com o tempo de execução?**

# Exercício 5

- ❑ Suponha que, ao executar um dado programa, um computador gaste 90% do seu tempo tratando um tipo especial de cálculo, e que seus fabricantes façam uma mudança que melhore o seu desempenho, naquele tipo de cálculo, por um fator de 10.
- ❑ Se o programa demorava, originalmente, 50 segundos para ser executado, qual será o seu tempo de execução depois da modificação?
- ❑ Qual é a aceleração do sistema novo em relação ao sistema antigo?

$$T_{\text{execução}_{\text{novo}}} = T_{\text{execução}_{\text{antigo}}} \times \left[ \text{Parcela}_{\text{não-usada}} + \frac{\text{Parcela}_{\text{usada}}}{\text{Aceleração}_{\text{usada}}} \right]$$

$$\text{Aceleração} = \frac{T_{\text{execução}_{\text{antigo}}}}{T_{\text{execução}_{\text{novo}}}} = \frac{1}{\left[ \text{Parcela}_{\text{não-usada}} + \frac{\text{Parcela}_{\text{usada}}}{\text{Aceleração}_{\text{usada}}} \right]}$$