

MC542

**Organização de Computadores
Teoria e Prática**

2006

Prof. Paulo Cesar Centoducatte

ducatte@ic.unicamp.br

www.ic.unicamp.br/~ducatte

MC542

Arquitetura de Computadores

Desempenho

**“Computer Organization and Design:
The Hardware/Software Interface” (Capítulo 1)**

Sumário

- Introdução
- Desempenho de Computadores
 - Tempo de resposta & *Throughput*
 - *Desempenho Relativo*
 - *Medida de Desempenho*
- *CPI - Cycles per Instructions*
- Componentes Básicos de Desempenho
- SPEC

Introdução

- Desempenho -> Como medir o desempenho ?

AVIÃO	PASSAGEIROS	AUTOMIA (milhas)	VELOCIDADE (mph)	THROUGHPUT (passag.Xveloc)
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.

Desempenho de Computadores

- Em computação:

- Um programa sendo executado em duas *worksations* 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.

Desempenho de Computadores

- 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 a sistemas que usam múltiplos processadores para tarefas distinta.

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

Desempenho de Computadores

- 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.

Desempenho de Computadores

- Para uma máquina X:
 - para uma determinada tarefa:
 - $\text{Desempenho}_X = (1 / \text{tempo de execução}_X)$
- Comparando 2 máquinas X e Y, se:
 - $\text{Desempenho}_X > \text{Desempenho}_Y$
 - $\text{Tempo de execução}_Y > \text{Tempo de execução}_X$

Desempenho de Computadores

- Desempenho relativo

- $(\text{Desempenho}_x / \text{Desempenho}_y)$

- $D_x / D_y = (\text{Tempo de execução}_y / \text{Tempo de execução}_x)$

- 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.**

Desempenho de Computadores

- Medida de Desempenho → tempo
 - Tempo de Execução → 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.)
 - *CPU time* = user CPU time + system CPU time (**geralmente só se considera user CPU time**)
 - Clock time → período do clock (clock cycle) → segundos
Ex.: 2nseg
 - Frequência do clock (clock rate) → Hz
Ex.: 500 MHz

Desempenho de Computadores

- Tempo de CPU

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

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

- Exemplo

- Um programa roda em 10 seg. na máquina A, cuja freqüência de seu clock é de 400 MHz. Uma máquina B, a ser projetada, tem que rodar este programa em 6 seg. Que acréscimo na freqüência 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.

Desempenho de Computadores

- Solução

- A → 10 seg. → 400 MHz → k períodos
- B → 6 seg. → ? → 1.2 k períodos

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

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

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

Desempenho de Computadores

- 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:

	t_{ck}	CPI
• Máquina A	1 ns	2.0
• Máquina B	2 ns	1.2
- Qual a máquina mais rápida ? E quanto ?

Desempenho de Computadores

Solução:

I = n. de instruções do programa

$$T_{\text{cpuA}} = \frac{\text{Tempo da CPU}}{I \times 2.0 \times 1n} = 2.0n \times I \quad \rightarrow + \textit{rápida}$$

↓
Períodos de clock para o programa

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

$$\begin{aligned} (\text{CPU performance A} / \text{CPU performance B}) &= (2.4n \times I / 2.0n \times I) \\ &= 1.2 \quad \rightarrow 1.2 \textit{ vezes} \\ &\quad + \textit{rápida} \end{aligned}$$

Componentes Básicos de Desempenho

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

$$\begin{array}{c}
 \text{Períodos de clock da CPU} = \sum_{i=1}^n (\text{CPI}_i \times C_i) \\
 \begin{array}{ccc}
 \downarrow & & \downarrow \\
 & & \text{n. de instruções da classe } i \\
 \downarrow & & \downarrow \\
 \text{média do n. de períodos para instruções da classe } i & &
 \end{array}
 \end{array}$$

n → n. de classes de instruções

Exemplo

Um projetista de compiladores está tentando decidir 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 ?

Componentes Básicos de Desempenho

Solução

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

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

$$\text{CPU}_{\text{períodos de clock}} = \text{CPU}_{\text{pc}} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$$

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

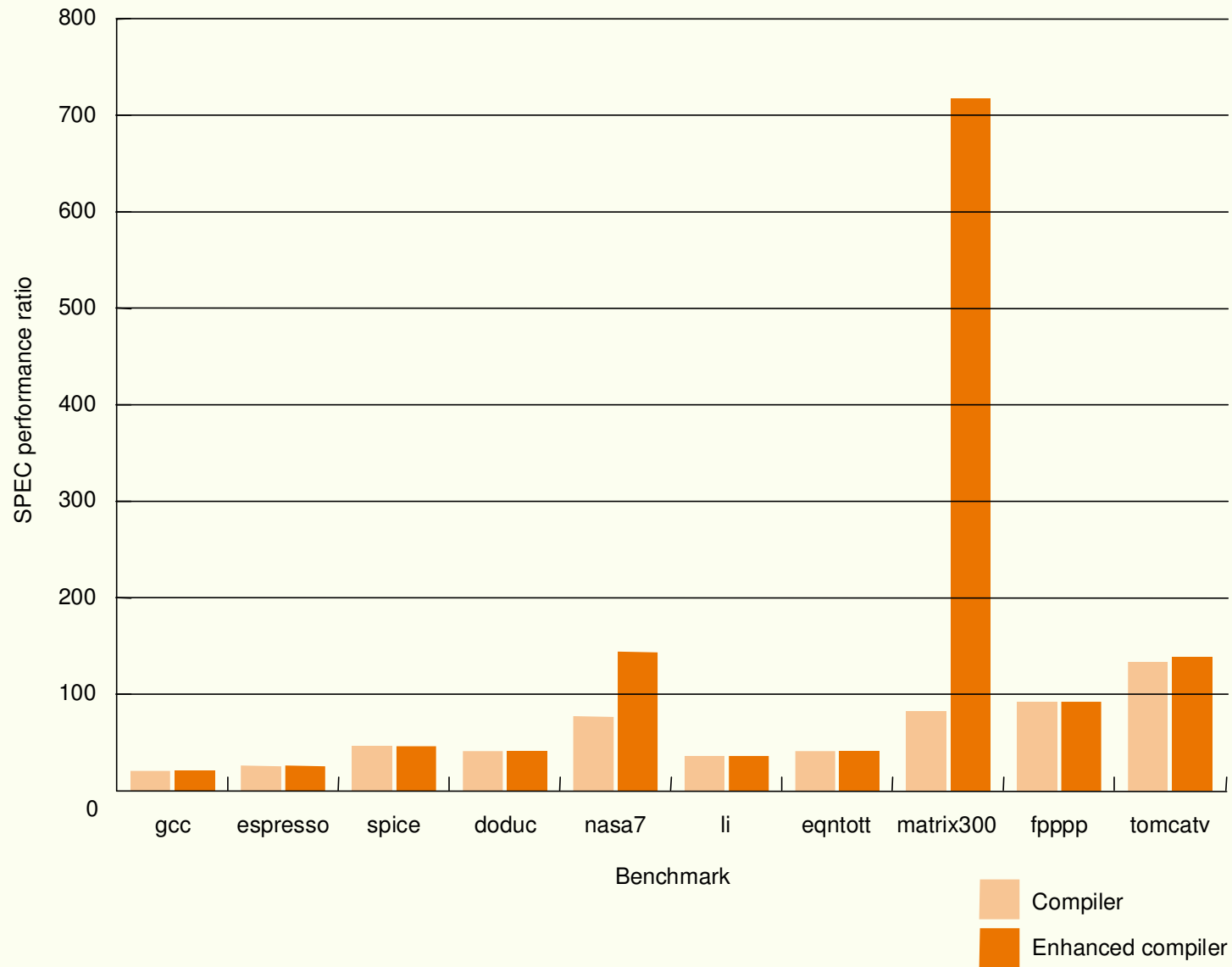
$$\text{CPU}_{\text{pc2}} = (4 \times 1) + (1 \times 2) + (1 \times 3) = 9 \text{ períodos} \rightarrow \textit{mais rápida}$$

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

$$\text{CPI}_1 = 10 / 5 = 2 \text{ (média de 2 períodos por instrução)}$$

$$\text{CPI}_2 = 9 / 6 = 1.5 \text{ (média de 1.5 períodos por instrução)}$$

SPEC



- Descrição do sistema utilizado na figura 2.3

Hardware	
Modelo	Powerstation 550
CPU	41.67 MHz POWER 4164
FPU	Integrado
Número de CPUs	1
Tamanho de cache por CPU	64K dados/ 8K instruções
Memória	64 MB
Disco	2400 MB SCSI
Interface de Rede	ND
Software	
Sistema Operacional	AIX v3.1.5
Compilador	AIX XL C/6000 v1.1.5
Outros software	AIX XL Fortran v2.2
Sistema de Arquivos	AIX
Firmware	ND
Sistema	
Parâmetros de Tuning	Nenhum
Background Load	Nenhum
Estado do Sistema	Multiusuário(somente 1 login)

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 :

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

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

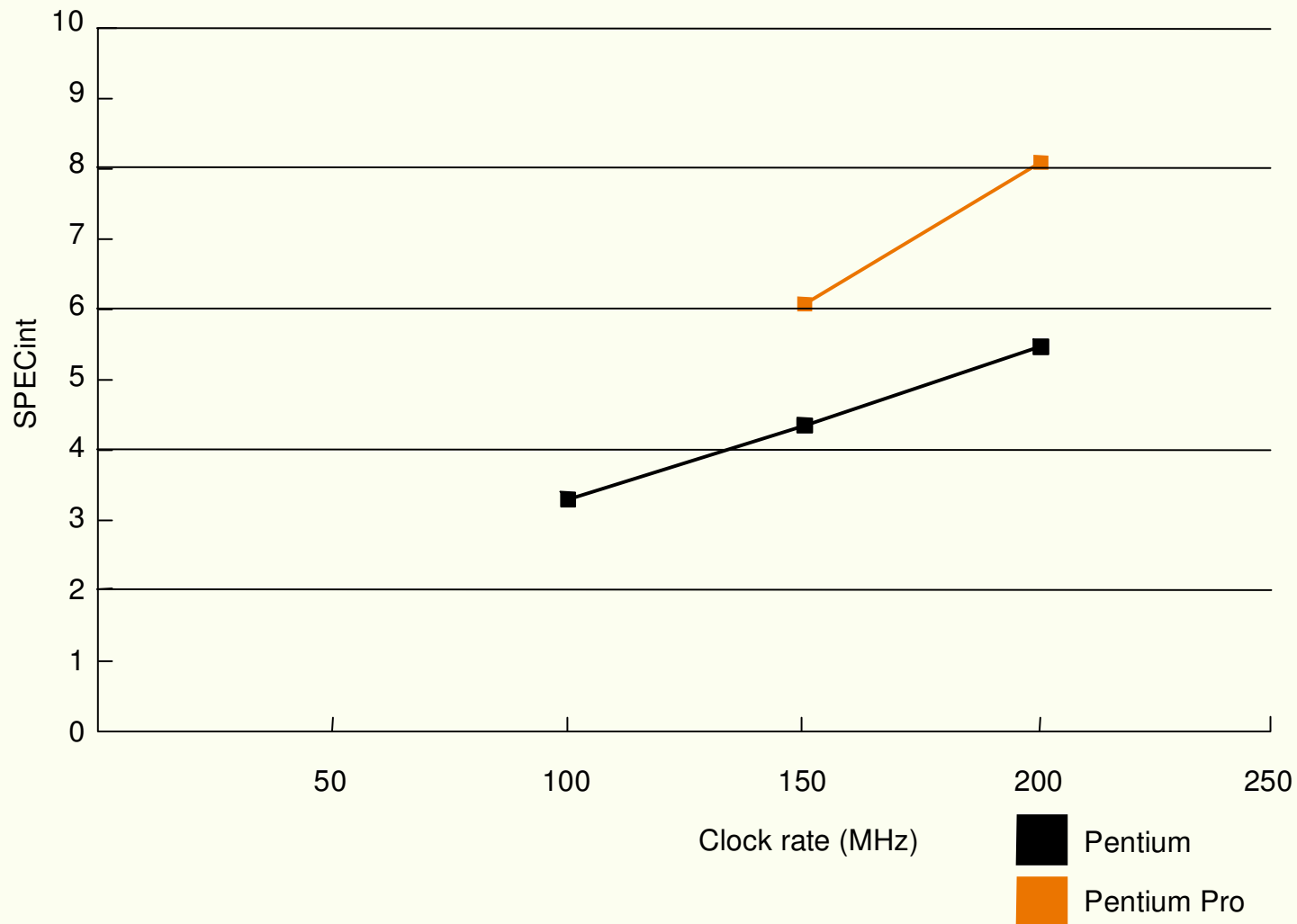
SPEC95 CPU benchmarks

Benchmark	Description
go	Artificial intelligence; plays the game of Go
m88ksim	Motorola 88K chip simulator; runs test program
gcc	The Gnu C compiler generating SPARC code
compress	Compresses and decompresses file in memory
li	Lisp Interpreter
jpeg	Graphic compression and decompression
perl	Manipulates strings and prime numbers in the special-purpose programming language Perl
vortex	A database program
tomcatv	A mesh generation program
swim	Shallow water model with 513 x 513 grid
su2cor	Quantum physics; Monte Carlo simulation
hydro2d	Astrophysics; Hydrodynamic Navier Stokes equations
mgrid	Multigrid solver in 3-D potential field
applu	Parabolic/elliptic partial differential equations
turb3d	Simulates isotropic, homogeneous turbulence in a cube
apsi	Solves problems regarding temperature, wind velocity, and distribution of pollutant
fpppp	Quantum chemistry
wave5	Plasma physics; electromagnetic particle simulation

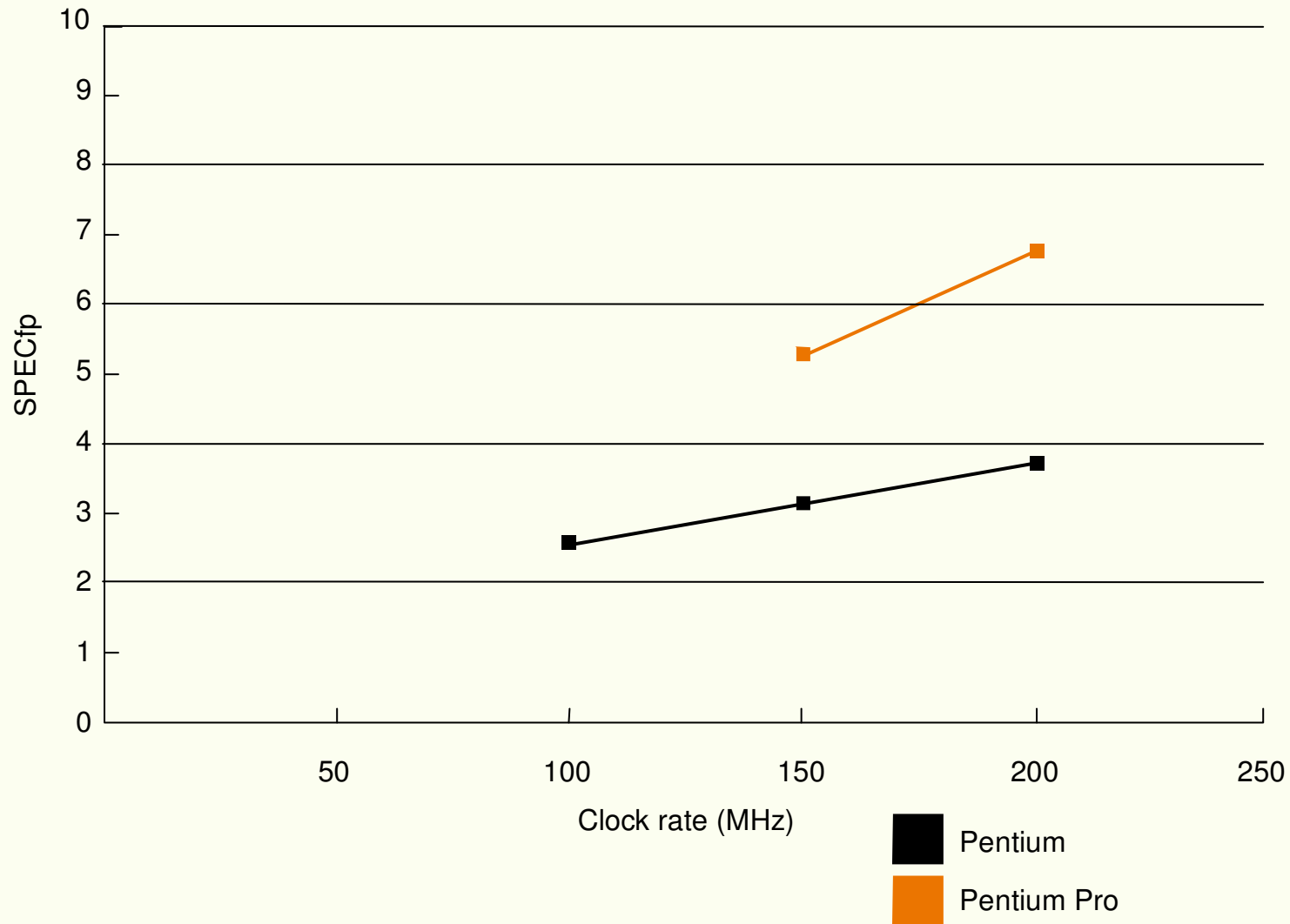
FIGURE 2.6 The SPEC95 CPU benchmarks. The 8 integer benchmarks in the top half of the table written in C, while the 10 floating-point benchmarks in the bottom half are written in Fortran 77. For more information on SPEC and on the SPEC benchmarks, see the link to the SPEC Web pages www.mkp.com/books_catalog/cod/links.htm.



Benchmark com SPECInt95 para Pentium e Pentium PRO com diferentes clocks



Benchmark com SPECfp95 para Pentium e Pentium PRO com diferentes clocks



MIPS

- Tentativa de padronizar uma medida de desempenho de um computador.
 - $MIPS = n. \text{ de instruções} / (\text{tempo de execução} \times 10^6)$
 - Problemas:
 - MIPS não leva em consideração o tipo de instrução (não se pode comparar computadores com Instruction Set diferentes)
 - MIPS varia entre programas num mesmo computador
 - MIPS pode variar inversamente com a performance

Exemplo

Máquina com 3 tipos de instruções e com CPI do exemplo da página 21

código do	n. de instruções por classe (bilhões)		
	A	B	C
compilador 1	5	1	1
compilador 2	10	1	1

frequência de clock = 500 MHz

Qual a seqüência de código que será executado mais rápido em MIPS e Tempo de Execução ?

$$\text{Tempo de execução} = \text{CPU}_{\text{pc}} / f_{\text{ck}}$$

$$\text{CPU}_{\text{pc}} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$$

$$\text{CPU}_{\text{pc1}} = (5 \times 1) + (1 \times 2) + (1 \times 3) \times 10^9 = 10 \times 10^9 \text{ períodos}$$

$$\text{CPU}_{\text{pc2}} = (10 \times 1) + (1 \times 2) + (1 \times 3) \times 10^9 = 15 \times 10^9 \text{ períodos}$$

$$\text{Tempo de execução}_1 = 10 \times 10^9 / 500 \times 10^6 = 20 \text{ s} \rightarrow$$

compilador 1 gera o código mais rápido

$$\text{Tempo de execução}_2 = 15 \times 10^9 / 500 \times 10^6 = 30 \text{ s}$$

$$\text{MIPS} = \text{n. instruções} / (\text{tempo execução} \times 10^6)$$

$$\text{MIPS}_1 = ((5 + 1 + 1) \times 10^9) / (20 \times 10^6) = 350$$

$$\text{MIPS}_2 = ((10 + 1 + 1) \times 10^9) / (30 \times 10^6) = 400 \rightarrow$$

compilador 2 tem a maior medida MIPS

Medidas?

- Como descrever em forma numérica o desempenho dos computadores?
- Quais ferramentas (ou qual ferramental) usar para realizar as medidas?

Métricas

Plane	DC to Paris	Speed	Passengers	Throughput (pmp)
Boeing 747	6.5 hours	610 mph	470	286,700
BAD/Sud Concorde	3 hours	1350 mph	132	178,200

- Tempo para executar uma tarefa (ExTime)
 - Execution time, response time, latency
- Tarefas por dia, hora, semana, segundo, ns, ... (Performance)
 - Throughput, bandwidth

Métricas Comparação

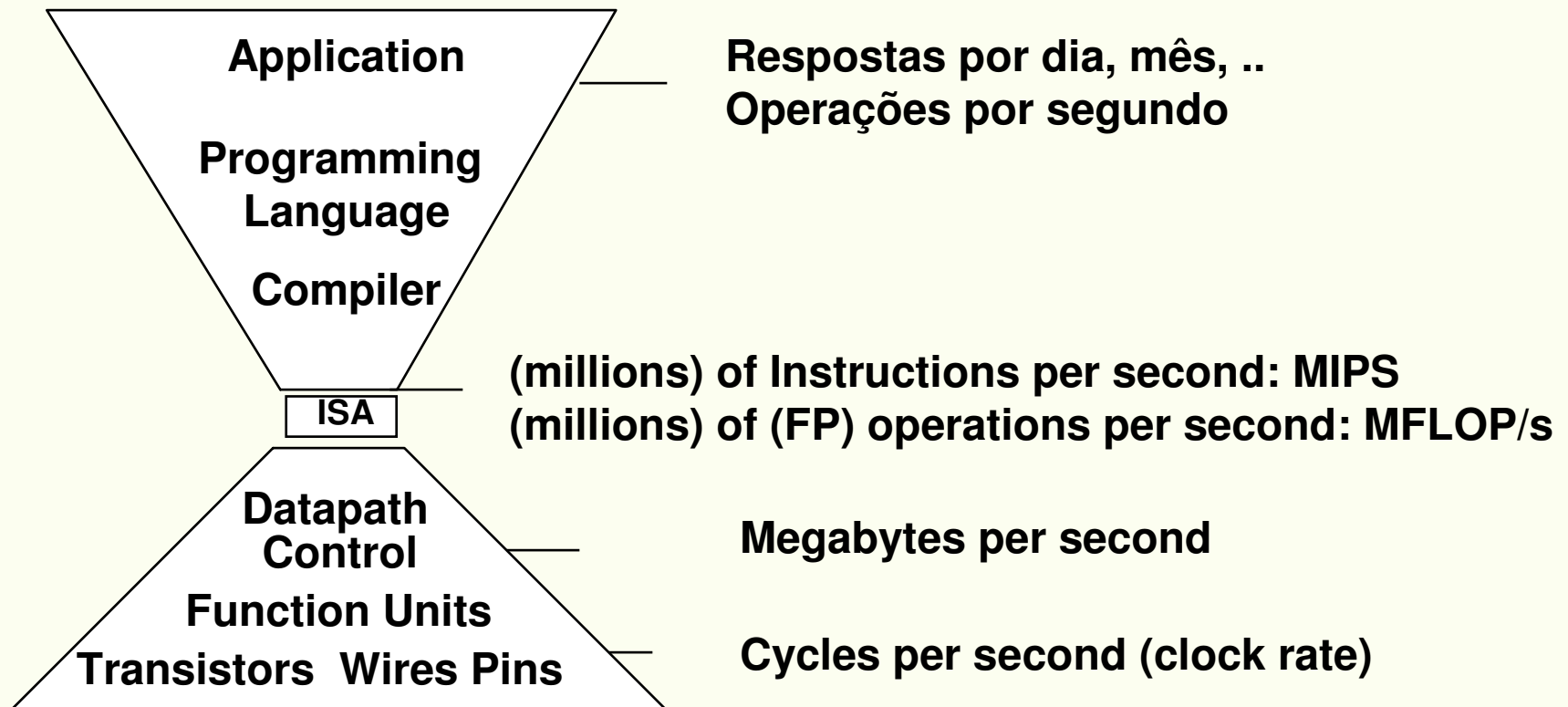
"X é n vezes mais rápido que Y" significa:

$$\frac{\text{ExTime}(Y)}{\text{ExTime}(X)} = \frac{\text{Performance}(X)}{\text{Performance}(Y)}$$

Velocidade do Concorde vs. Boeing 747

Throughput do Boeing 747 vs. Concorde

Métricas - Throughput



Métodos para Estimar o Desempenho

- Benchmarks, Traces, Mixes
- Hardware: Custo, delay, área, consumo de energia
- Simulação (vários níveis)
 - ISA, RT, Gate, Circuito
- Teoria das Filas
- Regras Práticas
- “Leis”/Princípios

Benchmarks

- **Aplicações Reais**
 - Compiladores, processadores de texto, ...
 - Problema de portabilidade, difícil medir o tempo de execução
- **Aplicação Modificadas**
 - Melhora a portabilidade, pode ser refinado para medir um certo aspecto de interesse (exp: tempo de cpu)
- **Kernels**
 - Usados para avaliar características específicas
 - Livermore Loops, Linpack
- **Toy Benchmarks**
 - 10 a 100 linhas de código, fácil de programar, avaliação inicial
- **Benchmarks Sintéticos**
 - Semelhantes aos Kernels
 - Whetstone, Dhrystone

Benchmarks

- Desktop
 - SPEC
- Servidores
 - SPEC
- Sistemas Embarcados
 - EEMBC (Embedded Microprocessor Benchmark Consortium)
(<http://www.eembc.org>)
 - Automotivo
 - Consumidor
 - Rede
 - Automação de Escritório
 - telecomunicações

Benchmarks

SPEC: System Performance Evaluation Cooperative
(<http://www.spec.org>)

- Primeira Versão - 1989
 - 10 programas ("SPECmarks")
- Segunda Versão - 1992
 - SPECInt92 (6 programas)
 - SPECfp92 (14 programas)
 - Compiler Flags: livre
- Terceira Versão - 1995
 - SPECint95 (8 programas)
 - SPECfp95 (10 programas)
 - SPECint_base95, SPECfp_base95
 - "benchmarks útil por 3 anos"
 - Compiler Flags: controladas

Benchmarks

SPEC CPU2000 - CINT2000

Programa	Linguagem	Finalidade
164.gzip	C	Compression
175.vpr	C	FPGA Circuit Placement and Routing
176.gcc	C	C Programming Language Compiler
181.mcf	C	Combinatorial Optimization
186.crafty	C	Game Playing: Chess
197.parser	C	Word Processing
252.eon	C++	Computer Visualization
253.perlbnk	C	PERL Programming Language
254.gap	C	Group Theory, Interpreter
255.vortex	C	Object-oriented Database
256.bzip2	C	Compression
300.twolf	C	Place and Route Simulator

<http://www.spec.org/osg/cpu2000/CINT2000/>

Benchmarks

SPEC CPU2000 - CFP2000

Program	Linguagem	Finalidade
168.wupwise	Fortran 77	Physics / Quantum Chromodynamics
171.swim	Fortran 77	Shallow Water Modeling
172.mgrid	Fortran 77	Multi-grid Solver: 3D Potential Field
173.applu	Fortran 77	Parabolic / Elliptic Differential Equations
177.mesa	C	3-D Graphics Library
178.galgel	Fortran 90	Computational Fluid Dynamics
179.art	C	Image Recognition / Neural Networks
183.earth	C	Seismic Wave Propagation Simulation
187.facerec	Fortran 90	Image Processing: Face Recognition
188.amm	C	Computational Chemistry
189.lucas	Fortran 90	Number Theory / Primality Testing
191.fma3d	Fortran 90	Finite-element Crash Simulation
200.sixtrack	Fortran 77	High Energy Physics Accelerator Design
301.apsi	Fortran 77	Meteorology: Pollutant Distribution

<http://www.spec.org/osg/cpu2000/CFP2000/>

Benchmarks - Exemplo de Resultado para SpecINT2000

<http://www.spec.org/osg/cpu2000/results/res2000q3/cpu2000-20000718-00168.asc>

Benchmarks	Base	Base	Base	Peak	Peak	Peak
	Ref Time	Run Time	Ratio	Ref Time	Run Time	Ratio
164.gzip	1400	277	505*	1400	270	518*
175.vpr	1400	419	334*	1400	417	336*
176.gcc	1100	275	399*	1100	272	405*
181.mcf	1800	621	290*	1800	619	291*
186.crafty	1000	191	522*	1000	191	523*
197.parser	1800	500	360*	1800	499	361*
252.eon	1300	267	486*	1300	267	486*
253.perlbnk	1800	302	596*	1800	302	596*
254.gap	1100	249	442*	1100	248	443*
255.vortex	1900	268	710*	1900	264	719*
256.bzip2	1500	389	386*	1500	375	400*
300.twolf	3000	784	382*	3000	776	387*
SPECint_base2000			438			
SPECint2000						442

Intel OR840(1 GHz
Pentium III processor)

Benchmarks

Como apresentar o Desempenho?

Gerentes gostam de números.

Técnicos querem mais:

- Reprodutibilidade - informações que permitam que o experimento seja repetido
- Consistência nos dados, ié se o experimento é repetido os dados devem ser compatíveis entre si

Como apresentar os Dados?

	Computador A	Computador B	Computador C
Programa P1 (secs)	1	10	20
Programa P2 (secs)	1000	100	20
Total Time (secs)	1001	110	40

Como Apresentar os Dados

- Média Aritmética (média aritmética ponderada)

$$\Sigma(T_i)/n \text{ or } \Sigma(W_i * T_i)$$

- Média Harmônica (média harmônica ponderada)

$$n/\Sigma(1/R_i) \text{ or } n/\Sigma(W_i/R_i)$$

- Média geométrica $(\prod T_j / N_j)^{1/n}$

- Tempo de execução normalizado (e.g., X vezes melhor que SPARCstation 10)

- Não use média aritmética para tempos de execução normalizado (o resultado, quando comparadas n máquinas, depende de qual máquina é usada como referência), **use média geométrica**

Como Apresentar os Dados

máquina	A	B
programa 1	10 => t1A	20 => t1B
programa 2	30 => t2A	5 => t2B

Média aritmética normalizada em A:

$$(t1A/t1A + t2A/t2A)/2 = 1 < (t1B/t1A + t2B/t2A)/2 = 13/12$$

Média aritmética normalizada em B:

$$(t1A/t1B + t2A/t2B)/2 = 13/4 > (t1B/t1B + t2B/t2B)/2 = 1$$

CONTRADIÇÃO

Média Geométrica :

$$((t1A * t2A)/(t1A * t2A))^{-.5} = 1 > ((t1B * t2B)/(t1A * t2A))^{-.5} = (1/3)^{-.5} \Rightarrow \textit{normalizado em A}$$

$$((t1A * t2A)/(t1B * t2B))^{-.5} = 3^{-.5} > ((t1B * t2B)/(t1B * t2B))^{-.5} = 1 \Rightarrow \textit{normalizado em B}$$