

2 . DESEMPENHO DE COMPUTADORES

- Performance -> Como medir o desempenho ?

Desempenho de um avião

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 formas diferentes, p. ex., velocidade, n. de passageiros, (n. passag. X veloc.), etc.

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.

- **Tempo de resposta & “Throughput”**

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

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

- 1. Mudar o processador por um mais rápido**
- 2. Adicionar mais um processador a sistemas que usam múltiplos processadores para tarefas distintas**

Resposta

Em 1. O tempo de resposta e o “throughput” crescem, enquanto em 2, o tempo de resposta continua o mesmo e o “throughput” cresce.

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

- **Para uma máquina X:**

Performance_X = (1 / tempo de execução_X); para uma determinada tarefa.

- Comparando 2 máquinas X e Y, se:

Performance_X > Performance_Y →

→ Tempo de execução_Y > Tempo de execução_X

- Performance relativa

(Performance_X / Performance_Y) = (Tempo de execução_Y / Tempo de execução_X)

Se X é n vezes mais rápida 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 = (Tempo de execução_B / Tempo de execução_A) = 1.5 →

→ A é 1.5 vezes mais rápido que B.

- Medida de Performance → 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*)

- Exemplo: comando `time` do UNIX

90.7u	12.9s	2:39	65%	
↓	↓	↓	↳	$((90.7+12.9)/159)=0.65$ →
			↓	→ % do <i>elapsed time</i> relativo
				ao <i>CPU time</i>
		↓		
		↓		<i>elapsed time</i> (159)
		↓		<i>system CPU time</i>
		↓		<i>user CPU time</i>

- *CPU performance* → *user CPU time*
- *Clock time* → período do *clock* (*clock cycle*) → segundos
Ex.: 2nseg
→ frequência do *clock* (*clock rate*) → Hz
Ex.: 500 MHz

- **Tempo de CPU**

$$\begin{aligned} T_{\text{CPU}}(\text{para um programa}) &= \\ &= \text{per\u00edodos de } \textit{clock} \text{ da CPU (para um} \\ &\quad \text{programa)} \times \text{per\u00edodo do } \textit{clock} \end{aligned}$$

$$\begin{aligned} T_{\text{CPU}}(\text{para um programa}) &= \\ &= \text{per\u00edodos de } \textit{clock} \text{ da CPU (para um} \\ &\quad \text{programa)} / \text{freq\u00fc\u00eancia do } \textit{clock} \end{aligned}$$

Exemplo

Um programa roda em 10 seg. na m\u00e1quina A, cuja freq\u00fc\u00eancia de *clock* \u00e9 de 400 MHz. Uma m\u00e1quina B, a ser projetada, tem que rodar este programa em 6 seg. Que acr\u00e9scimo na freq\u00fc\u00eancia de *clock* \u00e9 poss\u00edvel, sabendo-se que causar\u00e1 um acr\u00e9scimo de 1.2 vezes mais em n\u00famero de per\u00edodos de *clock* da m\u00e1quina A.

Solu\u00e7\u00e3o

$$\begin{aligned} A &\rightarrow 10 \text{ seg.} \rightarrow 400 \text{ MHz} \rightarrow k \text{ per\u00edodos} \\ B &\rightarrow 6 \text{ seg.} \rightarrow ? \rightarrow 1.2 k \text{ per\u00edodos} \end{aligned}$$

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

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

B tem que ter um *clock* com freq\u00fc\u00eancia duas vezes maior que A.

- n. de períodos da CPU (para um programa) =
= n. de instruções X n. médio de períodos por instrução (CPI)

$$T_{\text{CPU}}(\text{para um programa}) = \\ = (\text{n. de instruções X CPI}) / f_{\text{ck}}$$

$$T_{\text{CPU}}(\text{para um programa}) = \\ = (\text{n. de instruções X CPI}) \times t_{\text{ck}}$$

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 ?

Solução:

I = n. de instruções do programa

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

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

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

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 ?

Solução

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

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

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

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

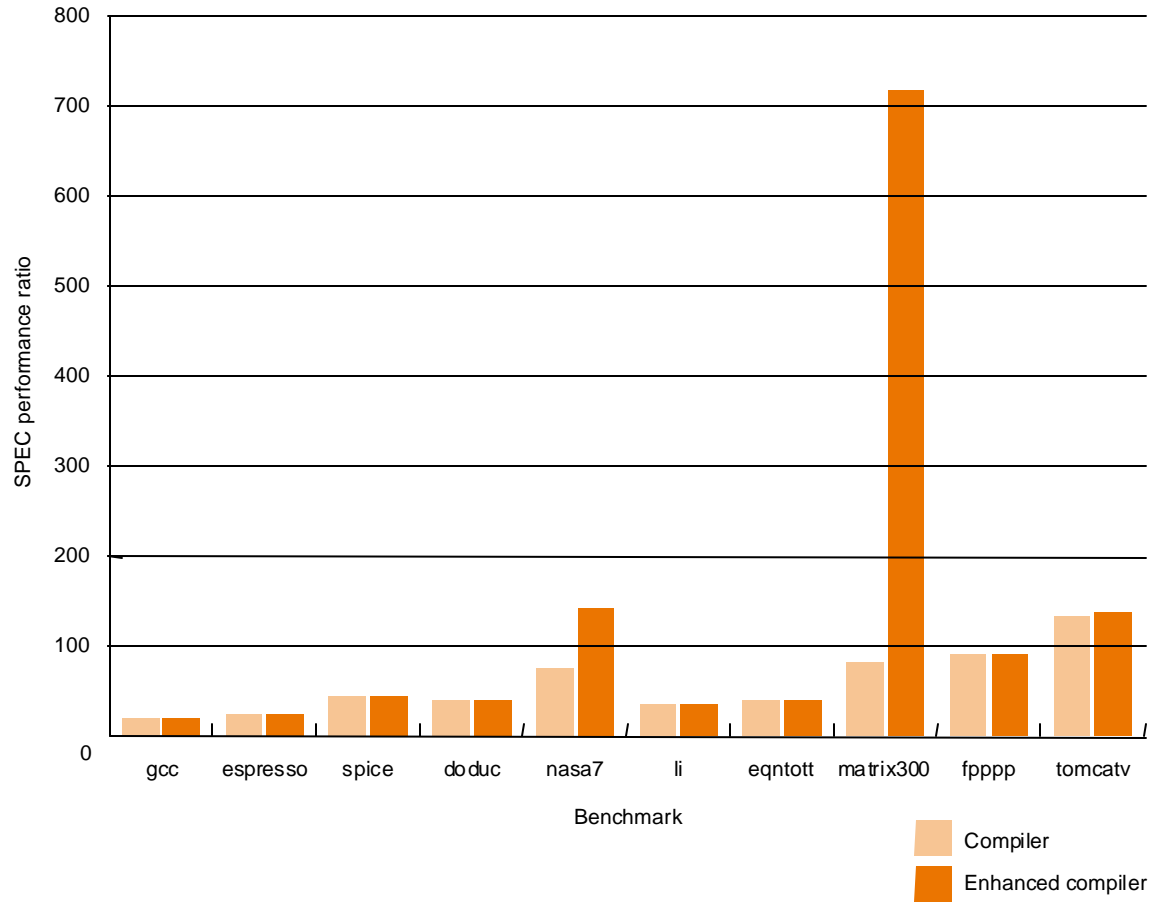
$$CPU_{\text{pc2}} = (4 \times 1) + (1 \times 2) + (1 \times 3) = 9 \text{ períodos} \rightarrow \textit{mais rápida}$$

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

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

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

- **SPEC**



- **Descrição do sistema utilizado na figura 2.3**

Hardware	
Model number	Powerstation 550
CPU	41.67-MHz POWER 4164
FPU	Integrated
Number of CPUs	1
Cache size per CPU	64K data/8K instruction
Memory	64 MB
Disk subsystem	2 400-MB SCSI
Network interface	NA
Software	
OS type and rev	AIX v3.1.5
Compiler rev	AIX XL C/6000 Ver. 1.1.5 AIX XL Fortran Ver. 2.2
Other software	None
File system type	AIX
Firmware level	NA
System	
Tuning parameters	None
Background load	None
System state	Multiuser (single-user login)

FIGURE 2.4 System description of the machine used to obtain the higher performance results in Figure 2.3. A footnote attached to the entry for the Fortran compiler states: "AIX XL Fortran Alpha Version 2.2 used for testing." Although no tuning parameters are indicated, additional footnotes describe a number of special flags passed to the compilers for the benchmarks.

Exemplo de utilização de um conjunto de programas para um *Benchmark*:

	Computador A	Computador B
Programa 1	1	10
Programa 2	1000	100
Total (secs)	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 !

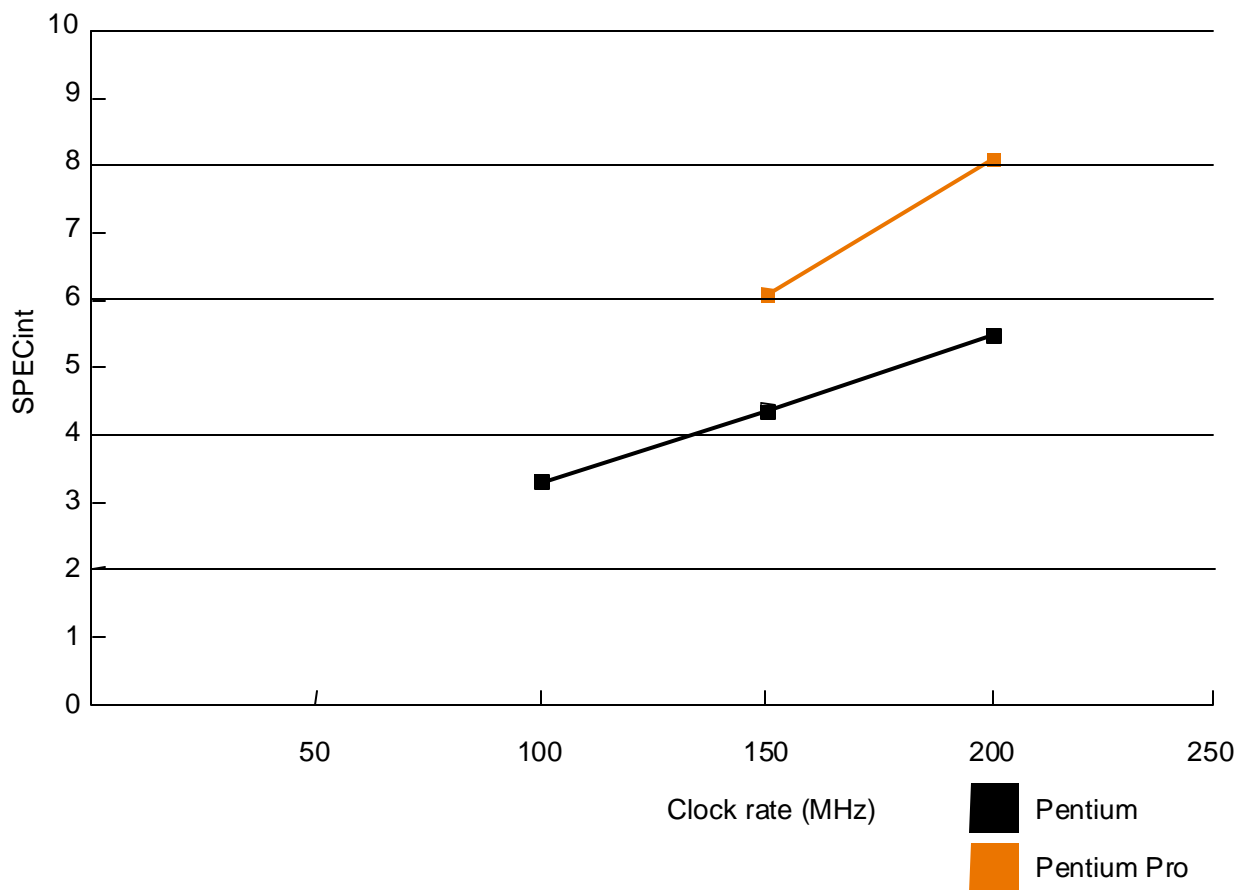
- **SPEC95 CPU Benchmarks**

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
ijpeg	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
fpccc	Quantum chemistry
wave5	Plasma physics; electromagnetic particle simulation

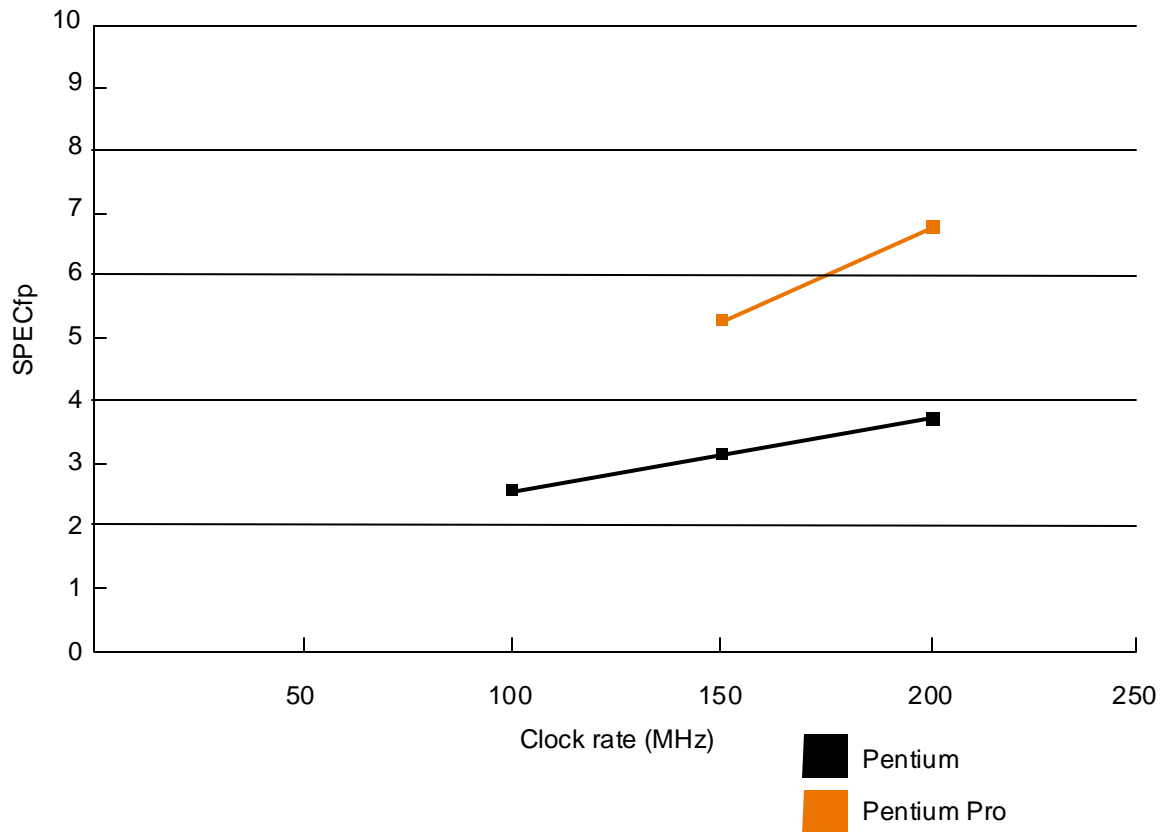


FIGURE 2.6 The SPEC95 CPU benchmarks. The 8 integer benchmarks in the top half of the table are written in C, while the 10 floating-point benchmarks in the bottom half are written in Fortran 77. For more information on SPEC and on the SPEC benchmarks, see the link to the SPEC Web pages www.mhp.com/books_catalog/cod/links.htm.

- **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.**

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

- **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

freqüência de *clock* = 500 MHz

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

Tempo de execução = CPU_{pc} / f_{ck}

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

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

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

Tempo de execução₁ = $10 \times 10^9 / 500 \times 10^6 = 20 \text{ s}$ → *compilador 1 gera o código mais rápido*

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

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

$$MIPS_1 = ((5 + 1 + 1) \times 10^9) / (20 \times 10^6) = 350$$

$$MIPS_2 = ((10 + 1 + 1) \times 10^9) / (30 \times 10^6) = 400 \rightarrow \text{compilador 2 tem a maior medida MIPS}$$