

PRESTAZIONI

Dal punto di vista UTENTE

TEMPO DI RISPOSTA

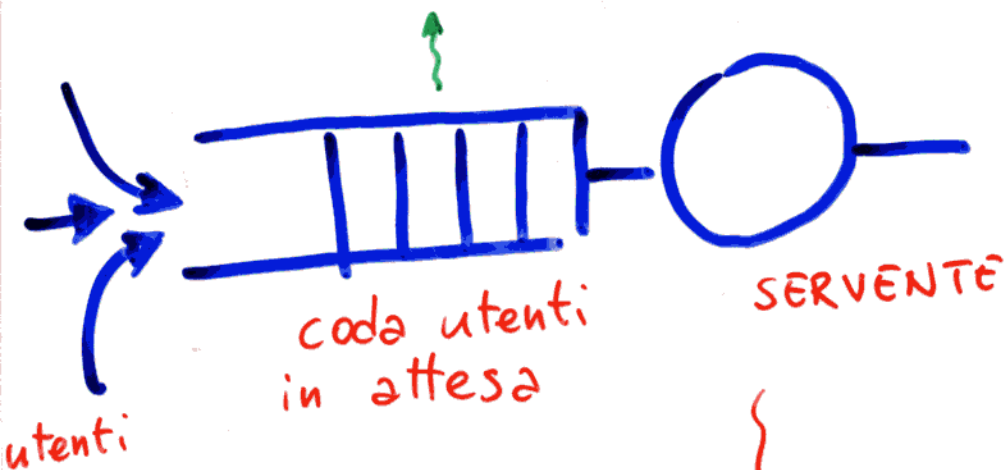
dal punto di vista SISTEMA

THROUGHPUT = n° di lavori (job) eseguiti nell'unità di tempo

TEMPO DI RISPOSTA } dipendono da:
THROUGHPUT

- velocità CPU
- dimensione CACHE
- " memoria di lavoro
- tempi di accesso del DISCO
- velocità dei bus
- n° di utenti nel sistema
- politiche di allocazione della CPU
- politiche di gestione della memoria
- ...

T_w : tempo attesa in coda



λ rate di arrivo

T_s tempo di servizio

$T_s = f$ (tipo servizio richiesto dall'utente, velocità esecuzione delle attività)

CASO CPU

Tempo di esecuzione di un programma = n° di cicli di clock della CPU per eseguire il programma \times durata del ciclo di clock

Programma \rightarrow sequenza di istruzioni

tipi di istruzioni

\rightarrow n° di clocks Per Instruction
CPI

tipi di istruzioni	LOAD	-----	CPI_L
	ADD	-----	CPI_A
	JMP	-----	CPI_J

Cicli di ck della CPU per eseguire un programma

$$= \sum_{i=1}^{N_{tipi}} CPI_i \cdot C_i$$

\nearrow
n° di istruzioni di tipo i eseguite nel programma

$$\text{Tempo di esecuz.} = \frac{\sum_{i=1}^{N_{\text{IPI}}} \text{CPI}_i \cdot C_i}{\text{freq. CK}} =$$

$$= \frac{\frac{N^{\circ} \text{istruz. eseg.}}{N^{\circ} \text{istruz. eseg.}} \sum_{i=1}^{N_{\text{IPI}}} \text{CPI}_i C_i}{\text{freq. CK}} =$$

$$= \frac{N^{\circ} \text{istruz. eseg.} \cdot \frac{\sum_{i=1}^{N_{\text{IPI}}} \text{CPI}_i C_i}{N^{\circ} \text{istruz. eseg.}}}{\text{freq. CK}} =$$

$$= \frac{N^{\circ} \text{istruz. eseg.} \times \text{CPI}}{\text{freq. CK}}$$

CPI dipende dal
programma di riferimento
scelto

CPI è un valore
medio

ESEMPIO: un progettista di compilatori sta cercando di decidere fra due sequenze di codice per una particolare macchina. I progettisti hw hanno fornito i seguenti dati:

Classe istruz.	CPI relativo
A	1
B	2
C	3

le due sequenze di codice richiedono

	A	B	C
1 ^a sequenza	2	1	2
2 ^a sequenza	4	1	1

Qual'è la sequenza più veloce?
Qual'è il CPI per ogni sequenza?

Cicli ck 1^a sequenza = 10

Cicli ck 2^a sequenza = 9

$$CPI_{1^{seq}} = \frac{\text{Cicli ck } 1^{a} \text{ sequenza}}{N^{\circ} \text{ di istruz. } 1^{a} \text{ seq}} = \frac{10}{5} = 2$$

$$CPI_{2^{seq}} = \frac{\text{Cicli ck } 2^{a} \text{ sequenza}}{N^{\circ} \text{ di istruz. } 2^{a} \text{ seq.}} = \frac{9}{6} = 1.5$$

MIPS: Mega Instructions Per Second

5

$$\text{MIPS} = \frac{\text{N}^\circ \text{ di istruz. di un programma} \cdot 10^{-6}}{\text{Tempo esec. del progr.}} =$$

$$= \frac{\text{N}^\circ \text{ istr. di un prog.} \cdot 10^{-6}}{\frac{\text{N}^\circ \text{ istruz di un prog.} \cdot \text{CPI}}{\text{freq. CK}}} = \frac{\text{freq. CK}}{\text{CPI}} \cdot 10^{-6}$$

non univocità di
CPI

$$\text{CPI} = \sum_{i=1}^{N_{\text{TIPI}}} \text{CPI}_i \cdot \frac{C_i}{N_{\text{istr. di un prog}}}$$

MIPS è collegato al programma
oltre che alla CPU

5

LIMITI NELL'USARE I MIPS COME CRITERIO DI CONFRONTO TRA DUE CPU:

1. MIPS specificano la frequenza di utilizzo delle istruzioni senza specificare il tipo delle istruzioni. Quindi non è possibile confrontare CPU con diversi set delle istruzioni
2. i MIPS non sono univoci per ogni singola CPU, dipendono dal programma che si usa
3. non è detto che per una stessa CPU a MIPS maggiori corrispondano tempi di esecuzione minori.

ESEMPIO ($CPI_A = 1$, $CPI_B = 2$, $CPI_C = 3$)

	classi istruz.		
	A	B	C
programma 1	5	1	1
programma 2	10	1	1

↖
n° di istruz.
in milioni
per ogni classe

$$CPI_1 = \frac{(5 \times 1 + 1 \times 2 + 1 \times 3) \cdot 10^6}{(5 + 1 + 1) \cdot 10^6} = \frac{10}{7} = 1,428$$

$$CPI_2 = \frac{(10 \times 1 + 1 \times 2 + 1 \times 3) \cdot 10^6}{(10 + 1 + 1) \cdot 10^6} = \frac{15}{12} = 1,25$$

ck di rif = 100 MHz.

$$MIPS_1 = \frac{100 \text{ MHz} \cdot 10^{-6}}{CPI_1} = 70$$

$$MIPS_2 = \frac{100 \text{ MHz} \cdot 10^{-6}}{CPI_2} = 80$$

$$\text{Tempo esec. 1} = \frac{N^\circ \text{ istruz. esec. 1}^\circ \text{ prop.} \times CPI}{100 \text{ MHz}} = 0,1 \text{ sec.}$$

$$\text{Tempo esec 2} = \frac{N^\circ \text{ istruz. esec. 2}^\circ \text{ prop} \times CPI}{100 \text{ MHz}} = 0,15 \text{ sec}$$

MFLOPS

Mega

Floating

Operations

Per

Second

$$\text{MFLOPS} = \frac{\text{N}^\circ \text{ di oper. in virgola mobile in un prog.} \cdot 10^{-6}}{\text{Tempo di esecuz. del prog.}}$$

il valore dipende:

- dal tipo di programma
- dal tipo delle operazioni in virgola mobile
- dalla velocità di esecuzione delle operazioni

↓
concludendo

indice non univoco

BENCHMARKS

per superare la non univocita' dei MIPS e MFLOPS



$$T_s = f \left(\begin{array}{l} \text{servizio} \\ \text{richiesto} \\ \text{dall'utente} \end{array}, \begin{array}{l} \text{velocita'} \\ \text{esecuzione} \\ \text{delle} \\ \text{attivit\`a} \end{array} \right)$$

CARICO

di LAVORO : insieme dei programmi che debbono essere eseguiti

BENCHMARKS : programmi caratterizzanti
un ambiente applicativo

però si trascura sempre il tempo
di attesa in coda - T_w

SPEC 95 - 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
ljpeg	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

C

FORTRAN
77
"floating
point"



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 at www.mkp.com/books_catalog/cod/links.htm.

SPEC = $\frac{\text{tempo di esecuzione del programma su SUN SPARC 10/40}}{\text{tempo di esecuzione del programma sul calcolatore di rifer.}}$

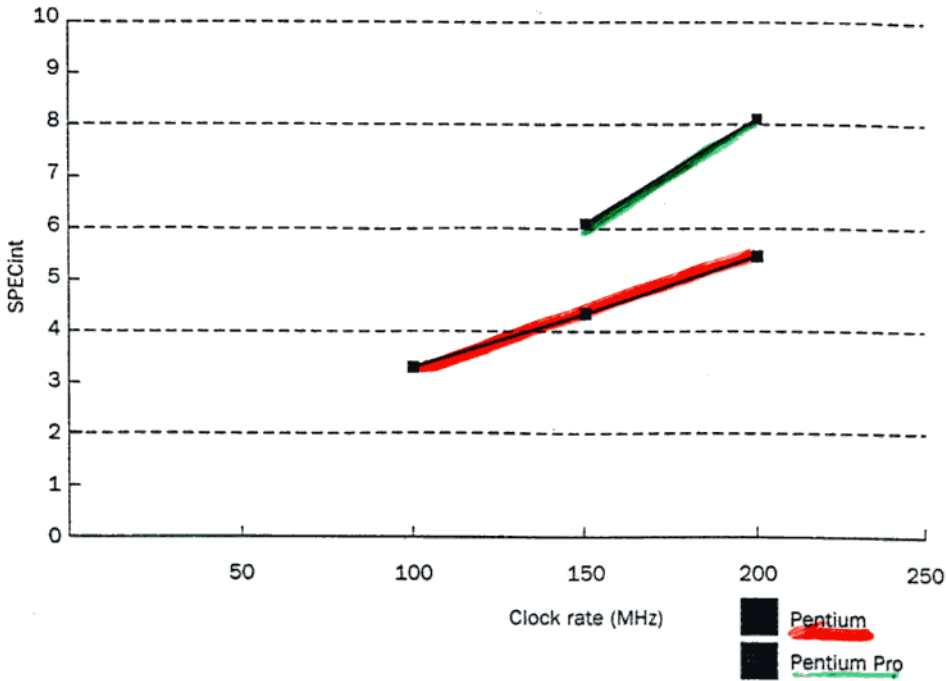


FIGURE 2.7 The SPECint95 ratings for the Pentium and Pentium Pro processors at different clock speeds. SPEC requires two sets of measurements: one that allows aggressive optimization with benchmark-specific switches, and one that allows only the standard optimization switches (called SPECint_base95). For the integer benchmarks on these processors, the results are the same. The link to these results can be found at www.mkp.com/books_catalog/cod/links.htm.

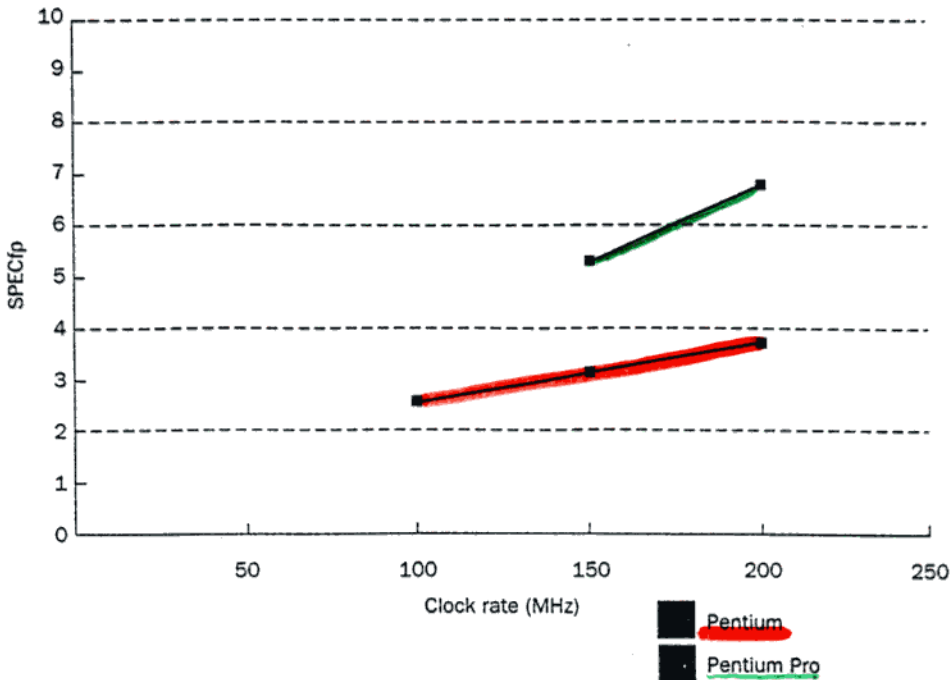


FIGURE 2.8 The SPECfp95 ratings for the Pentium and Pentium Pro processors at different clock speeds. These results can be found via the link to SPEC at www.mkp.com/books_cataog/cod/links.htm.



Domanda:

Se si velocizza di un fattore N la CPU e' ragionevole pensare che il tempo di esecuzione di un programma si riduca di un fattore N ?

$$\text{Speedup} = \frac{\text{tempo esecuz. DOPO miglior.}}{\text{tempo esecuz. PRIMA miglior.}}$$

Legge di Amdahl

tempo esecuzione
DOPO miglioramento =

$$= \frac{\text{tempo esecuzione influenz. dal miglior}}{\text{ammontare del miglior.}} +$$

+ tempo esecuzione NON influenzato
dal miglioramento

PROBLEMA: si supponga che un programma venga eseguito in 100 secondi su una macchina e che 80 di questi sono impegnati in addizioni. DI QUANTO BISOGNA MIGLIORARE LA VELOCITA' DI ESECUZIONE DELL'ADDIZIONATORE SE SI VUOLE CHE IL PROGRAMMA VENGA ESEGUITO 5 VOLTE PIU' RAPIDAMENTE?

$$\text{Tempo esec.} \\ \text{DO 20 miglior} = \frac{\text{tempo esec. influent. del migli.}}{\text{ammontare del miglior.}} + \text{tempo NON influent.}$$

$$20 \text{ sec} = \frac{80}{n} \text{ sec} + 20 \text{ sec}$$



$$n = \infty$$