Calcolatori Elettronici A
a.a. 2008/2009

PRESTAZIONI DEL CALCOLATORE

Massimiliano Giacomin

Due dimensioni

- ***Tempo di risposta*** (o ***tempo di esecuzione***):
  il tempo totale impiegato per eseguire un task (include il tempo CPU, il tempo per I/O, dischi, accesso alla memoria, overhead sistema operativo)
- ***Throughput***
  quantità di lavoro eseguita nell'unità di tempo

Esempio

- Processore più veloce $\Rightarrow \downarrow$ tempo di risposta e $\uparrow$ throughput
- Da uno a due processori $\Rightarrow \uparrow$ throughput,
  mentre $\downarrow$ tempo di risposta
  solo se carico > throughput

(comunque, i due fattori sono collegati…)

Importanza per diverse applicazioni e classi di computer

- Desktop computer:
  - tempo di risposta
- Server:
  - throughput e tempo di risposta
- Embedded computer:
  - vincoli real time sul tempo di risposta

hard real time: limite massimo fissato per eseguire un compito quando accade un determinato evento (es: ABS)

soft real time: tempo di risposta medio, oppure limite non superato almeno per una certa frazione degli eventi (es: DVD)

Focus su: tempo di esecuzione

- Per una macchina X (e un programma P) vale la relazione

$$\text{prestazioni}_X = \frac{1}{\text{tempo di esecuzione}_X}$$

- Se per due macchine X e Y, le prestazioni di X sono migliori di Y (sullo stesso programma P) si ha che

$$\text{prestazioni}_X > \text{prestazioni}_Y$$

$$\frac{1}{\text{tempo di esecuzione}_X} > \frac{1}{\text{tempo di esecuzione}_Y}$$

$$\text{tempo di esecuzione}_Y > \text{tempo di esecuzione}_X$$

Relazione fra le prestazioni di 2 macchine

- Se la macchina X è *n* volte più veloce della macchina Y si scriverà:

$$\frac{\text{prestazioni}_X}{\text{prestazioni}_Y} = n$$

- E quindi…

$$\frac{\text{prestazioni}_X}{\text{prestazioni}_Y} = \frac{\text{tempo di esecuzione}_Y}{\text{tempo di esecuzione}_X} = n$$

(diciamo anche che Y è *n* volte più lenta)

Fattori che influenzano le prestazioni

Per un dato programma, le prestazioni dipendono da:

- Algoritmo
- Linguaggio di programmazione
- Compilatore
- Sistema operativo
- Dispositivi di I/O
- La memoria
- Il processore

FOCUS SU: PROCESSORE

Tempo di CPU vs. tempo di risposta

Tempo di CPU: tempo speso dal processore per eseguire un dato programma, senza considerare il tempo speso per I/O o per altri programmi

Tempo di risposta: tempo percepito dall'utente

Noi: consideriamo il tempo di CPU, che in realtà include

- user CPU time (per il programma)
- system CPU time (per il sistema operativo)

Si considera un generico programma, senza distinzione utente vs. sistema operativo

Tempo di CPU: relazioni

$$\text{tempo di CPU relativo a un programma} = \text{cicli di clock della CPU relativi al programma} \times \text{periodo di ciclo del clock}$$

$$\text{tempo di CPU relativo a un programma} = \frac{\text{cicli di clock della CPU relativi al programma}}{\text{frequenza di clock}}$$

Ma i cicli di clock dipendono:

- dal numero delle istruzioni
- da quanti cicli di clock sono richiesti dalle istruzioni

CPI: clock Cycles Per Instruction

CPI (clock per istruzione):

il ***numero medio di cicli di clock per istruzione***,

calcolato come la media del numero di cicli di clock che le diverse istruzioni di un programma richiedono per essere completate

$$\text{cicli di clock della CPU} = \text{numero di istruzioni nel programma} \times \text{CPI}$$

$$T_{CPU} = \text{numero\_istruzioni} * \underbrace{CPI * T_{clock}}_{\text{Tempo medio di esecuzione per istruzione}}$$

$$= \frac{\text{numero di istruzioni} \times CPI}{f_{clock}}$$

Esempio

- Siano date due implementazioni diverse dello stesso set di istruzioni
- Il processore A ha una durata di ciclo di clock pari a 1 ns e un CPI pari a 2 per un determinato programma
- Il processore B ha una durata del ciclo di clock pari a 2 ns e un CPI pari a 1,2 per lo stesso programma
- Quale dei due è il più veloce? (sia I il numero di istruzioni del programma)

$$\text{cicli di clock della CPU}_A = I \times 2$$

$$\text{cicli di clock della CPU}_B = I \times 1{,}2$$

tempo di CPU$_A$ = cicli di clock della CPU$_A$ $\times$ periodo del ciclo di clock$_A$ = $I \times 2 \times 1$ ns = 2 I ns

tempo di CPU$_B$ = $I \times 1{,}2 \times 2$ ns = 2,4 I ns

$$\frac{\text{Prestazioni della CPU}_A}{\text{Prestazioni della CPU}_B} = \frac{\text{Tempo di esecuzione}_B}{\text{Tempo di esecuzione}_A} = \frac{2{,}4 \text{ I ns}}{2 \text{ I ns}} = \mathbf{1{,}2}$$

$\Rightarrow$ la CPU di A è 1,2 volte più veloce

Esercizio

Si supponga di disporre di due macchine: M_1 con frequenza di clock pari a 200 MHz e M_2 con frequenza di clock pari a 300 MHz; e si consideri un programma tale che:

	su M_1	su M_2
Tempo	10 sec	5 sec
Istruzioni eseguite	200×10^6	160×10^6

Trovare i CPI per eseguire il programma su ciascuna macchina

$$T_{M1} = I_{M1} \times CPI_{M1} / f_{M1} = 10 \text{ sec} = \frac{200 \cdot 10^6 \cdot CPI_{M1}}{200 \cdot 10^6 \text{ cicli/sec}} \qquad \mathbf{CPI_{M1}} = 10 \text{ cicli}$$

$$T_{M2} = I_{M2} \times CPI_{M2} / f_{M2} = 5 \text{ sec} = \frac{160 \cdot 10^6 \cdot CPI_{M2}}{300 \cdot 10^6 \text{ cicli/sec}} \qquad \mathbf{CPI_{M2}} = 9{,}375 \text{ cicli}$$

(durata ciclo $clock_{M1} = 1 / (200 \times 10^6 \text{ cicli/sec}) = 5$ ns
durata ciclo $clock_{M2} = 1 / (300 \times 10^6 \text{ cicli/sec}) = 3{,}\overline{3}$ ns)

Espressione dei CPI

- Si è visto che nel confronto tra processori che implementano la stessa ISA il numero di istruzioni non conta
- CPI può rappresentare il numero medio di cicli di clock delle istruzioni riferito a un programma ma più in generale al "carico di lavoro" previsto (***instruction mix***)

$$\text{CPI} = \frac{\sum_{i=1}^{n} (\text{CPI}_i \text{C}_i)}{I} \quad (\text{C}i\text{: numero di istruzioni i eseguite})$$

$$= \sum_{i=1}^{n} (f_i \, \text{CPI}_i)$$

dove f_i è la frequenza delle istruzioni della classe i e CPI_i è il numero di cicli per le istruzioni della classe i, mentre n è il numero delle classi

Esercizio

I progettisti hardware di un certo calcolatore hanno fornito le seguenti informazioni

Classe di istruzioni	CPI
A	1
B	2
C	3

Un progettista di compilatori deve scegliere tra due sequenze di codice

Sequenza di codice	Numero di istruzioni per ciascuna classe		
	A	B	C
1	2	1	2
2	4	1	1

Quale sequenza di codice causa l'esecuzione del maggior numero di istruzioni?
Quale viene eseguita più velocemente? Qual è il CPI per ciascuna delle sequenze?

Soluzione

- Sequenza 1: 2 + 1 + 2 = 5 istruzioni
  Sequenza 2: 4 + 1 + 1 = 6 istruzioni
  $\rightarrow$ Il numero minore di istruzioni si ottiene con la sequenza 1

- Usando la formula: cicli di clock della CPU = $\sum_{i=1}^{n} (\mathrm{CPI}_i \mathrm{C}_i)$

  Cicli di clock della $\mathrm{CPU}_1 = (2 \cdot 1)+(1 \cdot 2)+(2 \cdot 3)= 2 + 2 + 6 = 10$ cicli
  Cicli di clock della $\mathrm{CPU}_2 = (4 \cdot 1)+(1 \cdot 2)+(1 \cdot 3)= 4 + 2 + 3 = 9$ cicli
  $\rightarrow$ La sequenza 2 è più veloce

- Dato che: CPI = cicli di clock della CPU / numero di istruzioni

  CPI_1 = cicli di clock della CPU_1 / numero di $\mathrm{istruzioni}_1 = 10/5 = 2$
  CPI_2 = cicli di clock della CPU_2 / numero di $\mathrm{istruzioni}_2 = 9/6 = 1{,}5$

Fattori che influenzano il tempo di CPU

- Algoritmo: #istruzioni, CPI

  (es: se usa molti numeri FP, CPI più alto)
- Linguaggio di programmazione: #istruzioni, CPI

  (le caratteristiche del linguaggio influiscono sul numero e tipo di istruzioni macchina)
- Compilatore: #istruzioni, CPI

  (compilatori più o meno efficienti, possono tener conto delle caratteristiche del processore)
- ISA: #istruzioni, CPI, T_{clock}

  (RISC vs. CISC)
- Implementazione del processore (fissato ISA): CPI e T_{clock}

Valutazione di prestazioni del calcolatore attraverso benchmark

- ***Benchmark***: insieme di programmi scelti per misurare le prestazioni (approssimano il carico di lavoro di una classe di utenti)
- ***SPEC*** (System Performance Evaluation Corporation): definisce un insieme di benchmark standard per specifici ambienti:
  - orientati alla performance della CPU
  - grafica
  - object-oriented computing
  - web server
  - ecc. ecc.

  Includono regole precise per la reportistica: i report devono garantire la riproducibilità dei risultati (informazioni su processore, cache, memoria, sistema operativo, compilatore, ecc. ecc.)

Valutazione di prestazioni del calcolatore attraverso benchmark

- Valutazione:

  media pesata dei tempi di esecuzione P_i rispetto alla frequenza prevista di utilizzo f_i
- Problemi indotti dall'uso di benchmark:

  ottimizzazioni di particolari sequenze di istruzioni solo perché appaiono nei benchmark (es: opzioni di compilazione rivolte ai benchmark: ma cosa succede per le applicazioni generiche?)

Un modo scorretto (e old-style) di valutare le prestazioni: i MIPS

MIPS (milioni di istruzioni al secondo):

$$\text{MIPS} = \frac{\text{numero di istruzioni}}{\text{tempo di esecuzione} \times 10^6}$$

Evidenze del fatto che i MIPS non sono adeguati:

- "quante istruzioni al secondo" non è significativo, se non si sa che cosa fanno
- un programma più lento può avere MIPS più elevati! (per esempio, per aumentare i MIPS potrei inserire *nop*!!!)
- è impossibile dire che un calcolatore esegue *x* milioni di istruzioni al secondo, visto che i MIPS variano in base al programma eseguito

Legge di Amdahl: un principio di buon senso

Esempio

- Si supponga di aver migliorato una macchina in modo che le operazioni in virgola mobile vengano svolte 5 volte più velocemente
- Se prima del miglioramento il tempo di esecuzione di un dato benchmark era di 10 secondi quale sarà il tempo di esecuzione dopo il miglioramento supponendo che l'esecuzione delle operazioni in virgola mobile occupasse la metà dei 10 secondi?

$$\text{Tempo dopo il miglioramento} = \frac{5\,\text{secondi}}{5} + 5\,\text{secondi} = 6\,\text{secondi}$$

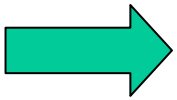
$$\text{Speedup} = \frac{\text{prestazioni dopo il miglioramento}}{\text{prestazioni prima del miglioramento}} = \frac{\text{tempo prima del miglioramento}}{\text{tempo dopo il miglioramento}} = \frac{10}{6} = 1{,}67$$

Le prestazioni migliorano solo di un fattore 1,67 e non di un fattore 5!!!

Legge di Amdahl: un principio di buon senso

- Se si migliora un aspetto/componente di una macchina, il possibile incremento delle prestazioni è **limitato** dall'ammontare dell'utilizzo dell'aspetto in questione

$$\text{Tempo di esecuzione dopo il miglioramento} = \frac{\text{tempo di esecuzione influenzato dal miglioramento}}{\text{ammontare del miglioramento}} + \text{tempo di esecuzione non influenzato}$$



Conviene ottimizzare i casi più frequenti!
(es. le istruzioni che vengono usate più frequentemente)