UNIVERSITÀ di VERONA

Dipartimento di INFORMATICA

PARCO Parallel Computing Lab

Misurazioni delle prestazioni in architetture parallele

Nicola Bombieri

Prestazione

valutazione

Utente:

- Per ridurre il *tempo di risposta.* Noto anche come tempo di esecuzione o latenza. Definito come il tempo trascorso dal inizio e il completamento del compito o del lavoro.

Sistema:

- Per aumentare il *portata* cioè, la quantità totale di lavoro svolto in un dato intervallo di tempo, a volte chiamato larghezza di banda.

Rapporto tra prestazioni

ØLa frase "X è più veloce di Y" viene utilizzata per dire che la risposta
il tempo o il tempo di esecuzione, per una determinata attività, è inferiore in X che in Y
Ø *"X è n volte più veloce di Y" significa:*

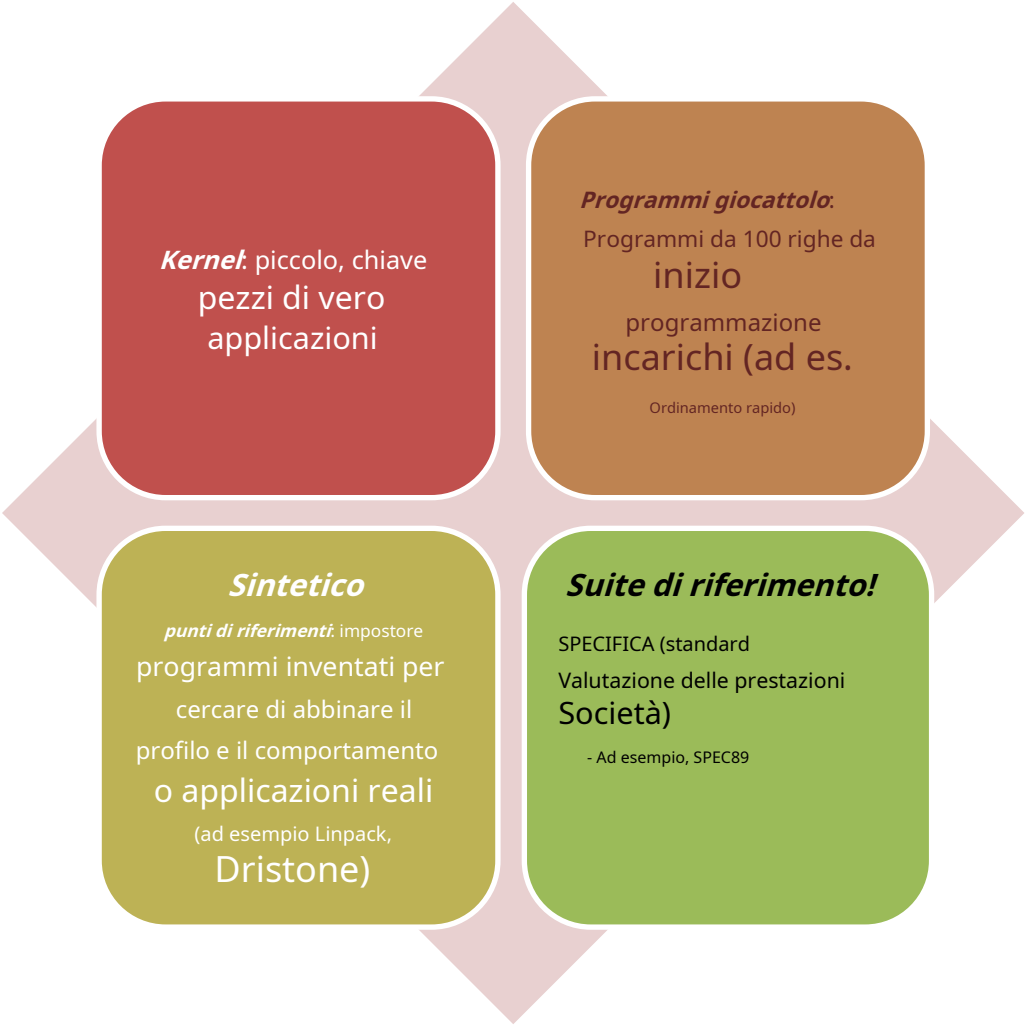
$$\frac{\text{Tempo di esecuzione}_{sì}}{\text{Tempo di esecuzione}_{X}} = n$$

Noi lo chiamiamo ***accelerare***!

ØExecution_time = reciproco della prestazione, quindi:

$$N = \frac{\text{Tempo di esecuzione}_{sì}}{\text{Tempo di esecuzione}_{X}} = \frac{\frac{1}{\text{Prestazione}_{sì}}}{\frac{1}{\text{Prestazione}_{X}}} = \frac{\text{Prestazione}_{X}}{\text{Prestazione}_{sì}}$$

Punti di riferimenti



Esempio: tabella delle prestazioni

Modello	Anno	Linpack(i)	Dristone (s)
VAX-11/780	1978	4,90	5.69
VAX-8600	1985	1.43	1.35
VAX-8550	1987	0,695	0,96

ØTempo di esecuzione in secondi per il benchmark Linpack
e per 10.000 iterazioni per il benchmark Dhrystone

Esempio (continua)

- Quanto è più veloce:
  - Utilizzando Linpack, il modello 8600 contro 780?
  - Quanto utilizzando Dhrystone?
  - Quanto costa il modello 8550 rispetto all'8600?
- Soluzione:

Principio quantitativo

- Per accelerare il caso più comune:
  - Dobbiamo concentrarci sul caso più frequente piuttosto che su quello quello più raro
  - Il caso più frequente è spesso il più semplice e può farlo essere accelerato più di quello non frequente.

- Amdahl la legge di:
  - Il miglioramento delle prestazioni che può essere ottenuto da l'utilizzo di qualsiasi modalità di esecuzione più veloce è limitato dalla frazione di tempo in cui tale modalità può essere applicata.

Esempio

Andando dal Nevada a Los Angeles – California attraverso le montagne della Sierra Nevada e attraversando il deserto.

- L'itinerario attraversa, in montagna, zone ecologicamente sensibili, da percorrere a piedi. La traversata delle montagne richiede 20 ore.
- Le ultime 200 miglia, invece, possono essere fatte con un mezzo molto veloce.

Esempio (continua)

- Alternative ultime 200 miglia:
  - A piedi con una media di 4 miglia/ora
  - In bicicletta con una media di 10 miglia/ora
  - Con un'auto di medie dimensioni con una media di 50 miglia/ora
  - Da una Ferrari Testarossa con una media di 120 miglia/ora
  - *Con un F-16*Combattimento Falcon supersonico aereo con una media di 600 miglia/ora
- Calcola l'accelerazione.

Esempio (continua)

Deserto (II parte del viaggio)			Totale	
Veicolo	**Ore**	**Accelerare**	**ore**	**Accelerare**
A piedi	50,00	1.0	70,00	1.0
In bicicletta	20.00	2.5	40,00	1.8
Auto di medie dimensioni	4.00	12.5	24.00	2.9
FerrariTestarossa	1.67	30.0	21.67	3.2
Aereo F-16	0,33	150,0	20.33	3.4

Legge di Amdahl

ØFrazione$_{migliorato}$(£1)

- ØLa frazione del tempo di esecuzione della macchina originale (o del codice originale) che può essere modificato per sfruttare i miglioramenti.
- ØNell'esempio precedente, la frazione era 50/70. Ø

Accelerare$_{migliorato}$(^{3}1)

- ØIl miglioramento ottenuto dalla modalità di esecuzione più veloce. Ø
- Nell'esempio precedente si trattava dell'accelerazione "nel deserto".

Legge di Amdahl

$$\text{Tempo di esecuzione}_{nuovo} = \text{Tempo di esecuzione}_{vecchio} * \left((1-\text{Frazione}_{migliorato}) + \frac{\text{Frazione}_{migliorato}}{\text{Accelerare}_{migliorato}}\right)$$

$$\text{Accelerare}_{globale} = \frac{\text{Execution\_time}_{vecchio}}{\text{Tempo di esecuzione}_{nuovo}} = \frac{1}{(1-\text{Frazione}_{migliorato}) + \frac{\text{Frazione}_{migliorato}}{\text{Accelerare}_{migliorato}}}$$

ØSe un miglioramento può essere utilizzato solo per una frazione dell'intero compito:

$$\text{Accelerare}_{globale} = \frac{1}{(1-\text{Frazione}_{migliorato}) + p\frac{\text{azione}_{migliorato}}{\text{Accelerare}_{migliorato}}} \pounds \frac{1}{(1-\text{Frazione}_{migliorato})}$$

Esercizio1

ØPrendere in considerazione:

- ØUn miglioramento 10 volte più veloce rispetto alla macchina originale (o codice),
- ØMa ciò può essere applicato solo per il 40% dei casi

ØE il guadagno totale? Ø

Soluzione:

Esercizio2

- Aggiornamento di una CPU:
  - Aumento della velocità di un fattore 5 (5 volte) (senza incidere sulle prestazioni degli I/O);
  - Costo: 5 volte superiore;
  - Supponiamo che la CPU possa essere utilizzata per il 50% del tempo, mentre il tempo rimanente viene dedicato alle operazioni di I/O;
  - Il costo della CPU è 1/3 del costo totale della macchina;
- È un investimento intelligente dal punto di vista del rapporto costo/prestazioni

punto di vista?

Esercizio 2 (soluzione)

Esercizio 3 (multiprocessore)

- Riscrittura di un programma da eseguire su:
  - MIMD con 100 processori;
  - L'obiettivo è ridurre i tempi di esecuzione del file originale (SISD) di 80 volte
- Quale frazione del programma originale può rimanere sequenziale?

Esercizio 3 (soluzione)

Esercizio 4(SIMD)

- Il calcolo parallelo è 20 volte più veloce del originale (sequenziale).
- La percentuale di parallelismo è la porzione di tempo spesi utilizzando la modalità di esecuzione SIMD.
  - Disegna un grafico per rappresentare il *accelerare* come percentuale del calcolo effettuato in modalità SIMD.
  - Quale percentuale di parallelismo è necessaria per avere a accelerazione pari a 2? Quale per raggiungere la metà della velocità massima?

Esercizio 4 (soluzione)

Esercizio 4bis

- La percentuale di parallelismo raggiungibile per an
  l'esempio è del 70%.
- Due alternative:
  - Raddoppiare la velocità della modalità SIMD (molto costosa)
  - Per aumentare la percentuale del parallelismo mettendo a fuoco
    sul compilatore.
- Quale aumento della percentuale di
  è necessario il parallelismo per ottenere lo stesso
  guadagno prestazionale?

Esercizio 4bis (soluzione)

Esercizio 5 (cache)

ØArchitettura basata sulla cache:

- ØMemoria cache 5 volte più veloce della memoria principale. Ø Cache utilizzata per il 90% del tempo.

ØCache più veloce rispetto a non cache? Ø

Soluzione:

Il tempo è la misura della performance

Il tempo di esecuzione è la misura delle prestazioni di un computer: The l'architettura che esegue la stessa quantità di tempo in meno tempo è la più veloce.

Tempo di risposta: rappresenta la latenza necessaria per eseguire un'attività includendo il disco accessi, accessi alla memoria, attività di I/O

Tempo della CPU:non comprende il tempo di attesa di un I/O o di esecuzione di un altre attività, corrisponde al tempo dell'utente + al tempo del sistema della CPU.

processoretime= Cicli di clock della CPU per un task * tempo di ciclo di clock

D**processore**tempo=$\frac{\text{Cicli di clock della CPU per un task}}{\text{frequenza dell'orologio}}$

Cicli di clock per istruzione (CPI)

$$IPC = \frac{\text{Cicli di clock della CPU per un task}}{\text{numero di istruzioni}}$$

Tempo CPU= n° istruzioni * CPI * durata del ciclo di clock

$$= \frac{\text{n° istruzioni} * \text{CPI}}{\text{frequenza dell'orologio}}$$

$T_{processore} = CI * IPC * T_{OROLOGIO} = (CI*CPI)/f_{OROLOGIO}$

Introducendo nella prima formula le unità, si può vedere il rapporto tra i diversi componenti:

$$\frac{\text{Istruzioni}}{\text{compito}} * \text{cicli} \frac{\text{dell'orologio}}{\text{cicli di clock delle istruzioni}} * \frac{\text{secondi}}{\text{dell'orologio}} = \frac{\text{secondi}}{\text{compito}} = \text{Tempo della CPU}$$

Il tempo della CPU dipende da 3 parametri

Ciclo di clock (o frequenza): Tecnologia HW e organizzazione

IPC: organizzazione e insieme di istruzioni

Numero di istruzioni: architettura del set di istruzioni (ISA) e la tecnologia del compilatore

Numero totale di cicli di clock

$$\text{Cicli di clock della CPU} = \sum_{io=1}^{N} (CPI_{io} * IO_{io})$$

IO_{io}=numero di volte in cui ho eseguito l'istruzione in un'attività

IPC_{io}=*media*numero di cicli di clock spesi per un generico istruzioni

È possibile scrivere il tempo della CPU come segue:

$$T_{processore} = \sum_{io=1}^{N} (IPC_{io} * IO_{io}) * T_{OROLOGIO}$$

Esercizio 6

Supponiamo di aver effettuato le seguenti misurazioni:

Frequenza delle operazioni FP (incluso FPSQR) = 25%

CPI medio delle operazioni FP = 4,0

CPI medio di altre istruzioni = 1,33 Frequenza dell'operazione FPSQR (radice quadrata FP) = 2% CPI di FPSQR = 20,0

Dobbiamo analizzare 2 alternative:

1) Ridurre il CPI di FPSQR a 2

2) Ridurre l'IPC medio di tutte le operazioni del PQ a 2,5.

Confronta le 2 alternative in termini di prestazioni complessive

Esercizio 6 (soluzione)

Esercizio 7

L'architettura inizialmente prevede solo istruzioni di caricamento e archiviazione per accedere alla memoria. Tutte le altre operazioni funzionano a registri (architettura L/S). Il riepilogo delle istruzioni è riportato in tabella.

	Frequenza	Cicli dell'orologio
ALLU	43%	1
Carico	21%	2
Negozio	12%	2
Ramo	24%	2

Supponiamo che il 25% delle operazioni ALU utilizzino *uno* operando appositamente caricato, che non è più utile alle operazioni successive. Proponiamo di aggiornare il set di istruzioni avendo in memoria uno degli operandi sorgente. Queste nuove operazioni reg-mem hanno un numero di cicli di clock pari a 2 e richiedono 3 cicli di clock per i rami, senza modificare il periodo di ciclo di clock.

Questa modifica può migliorare le prestazioni della CPU?

Esercizio 7 (soluzione)

Esercizio 8

Un'analisi relativa ai costrutti dei linguaggi di alto livello indica che le chiamate di procedura sono le operazioni più costose.

Esiste un'ottimizzazione che riduce le operazioni di caricamento e memorizzazione solitamente associate alle chiamate di procedura e ai relativi resi. Alcuni esperimenti con e senza questa ottimizzazione danno i seguenti risultati:

- Il ciclo di clock per la versione non ottimizzata è del 5% più veloce
- Il 30% delle istruzioni della versione non ottimizzata vengono caricate e memorizzate
- La versione ottimizzata esegue 1/3 del carico e memorizza meno di quella non ottimizzata (per le altre istruzioni il numero di istruzioni non cambia)
- Tutte le istruzioni (anche caricamento e memorizzazione) richiedono un ciclo di clock.

Qual è la soluzione più veloce?

Esercizio 8 (soluzione):

MIPS (GIPS)

MIPS = milioni di istruzioni al secondo
GIPS = miliardi di istruzioni al secondo

$$\text{MIPS} = \frac{\text{numero di istruzioni}}{\text{tempo di esecuzione} * 10^6} = \frac{\text{frequenza dell'orologio}}{\text{IPC} * 10^6}$$

$$\text{Tempo di esecuzione} = \frac{\text{numero di istruzioni}}{\text{MIPS} * 10^6}$$

Poiché MIPS è la frequenza delle operazioni per unità di tempo, le prestazioni per le macchine più veloci che hanno valori MIPS elevati possono essere specificate come l'inverso del tempo di esecuzione.
Il vantaggio di MIPS è che è semplice da capire:***le macchine più veloci hanno valori MIPS più alti!***

Problemi MIPS

• Il valore dipende dal set di istruzioni, quindi è difficile confrontare macchine con set di istruzioni diversi.
• Varia a seconda del programma considerato;
• Può variare in modo inversamente proporzionale rispetto alla prestazione!

Esempio:architettura con acceleratori specifici per le operazioni FP. Le istruzioni FP richiedono più cicli di clock rispetto alle istruzioni intere. Pertanto, i programmi che lavorano con un acceleratore esterno per FP invece che con routine SW per tali operazioni richiederanno meno tempo di calcolo ma avranno un MIPS inferiore.

Al contrario, l'implementazione SW delle istruzioni FP esegue istruzioni semplici, risultando in un MIPS più elevato. Tuttavia, esegue molte istruzioni e avrà un tempo di esecuzione molto elevato.

GFLOPS

GFLOPS = Miliardi di istruzioni FP al secondo

$$\text{GFLOPS} = \frac{\text{Numero di operazioni del PQ in un programma}}{\text{tempo di esecuzione} * 10^9}$$

GFLOPS consente di misurare le prestazioni relative alle operazioni FP quindi non può essere utilizzato in altri contesti diversi.

Essendo basato su operazioni piuttosto che su istruzioni, il FLOPS è concepito per essere un buon elemento di confronto tra diverse architetture. L'ipotesi è che lo stesso programma, lavorando su architetture diverse, eseguirebbe un numero diverso di istruzioni, ma *lo stesso numero di operazioni del PQ.*

- Le operazioni FP non sono coerenti tra le diverse architetture;
- GFLOPS cambia cambiando il rapporto tra operazioni intere e FP ma anche cambiando il mix tra operazioni FP veloci e lente!

GFLOPS normalizzati

Ad ogni operazione FP è associato un bias che rappresenta il peso:

Operazioni reali	**Pregiudizio**
AGGIUNGI, SOTTO, CONFRONTA,	1
MULTI DIVIDE, SQRT	4
ESP, PECCATO, ...	8

Si calcola che un frammento di codice avente un'operazione ADD, una DIVIDE e un'operazione SIN abbia 13 operazioni FP normalizzate.

Esercizio 9

Il programma Spice viene eseguito su una DECstation 3100 in 0,094 secondi. Nella tabella è riportato il numero di operazioni FP eseguite dal programma.

Qual è il GFLOPS del programma? E i GFLOPS normalizzati?

aggiungi D	25.999.440
sottoD	18.266.439
mulD	33.880.810
divD	15.682.333
rispetto	9.745.930
negD	2.617.846
absD	2.195.930
convertireD	1.581.450
TOTALE	**109.970.178**

Soluzione: