

OTTIMIZZAZIONE DEL PROGRAMMA

Modificare il programma in modo che la sua esecuzione minimizzi l'uso delle risorse disponibili sul calcolatore, massimizzando le prestazioni e senza alcun impatto (negativo) sulla correttezza

PERCHE' ?

- Mi permette di eseguire calcoli su una quantità maggiore di dati
- Eseguire concorrentemente un numero maggiore di software
- Usare meno energia
- ...

COME SI OTTIMIZZA UN PROGRAMMA ?

- 1) Come posso valutare l'impatto che ha una specifica ottimizzazione sul programma ?
 - ▶ Come misuro le prestazioni ?
- 2) Come posso scegliere le parti di un programma che posso/voglio ottimizzare ?
 - ▶ Come identifica in un programma i "punti caldi" (hot spot) meritevoli di attenzione.

METRICHE PRESTAZIONALI

LATENZA : tempo necessario per eseguire una operazione

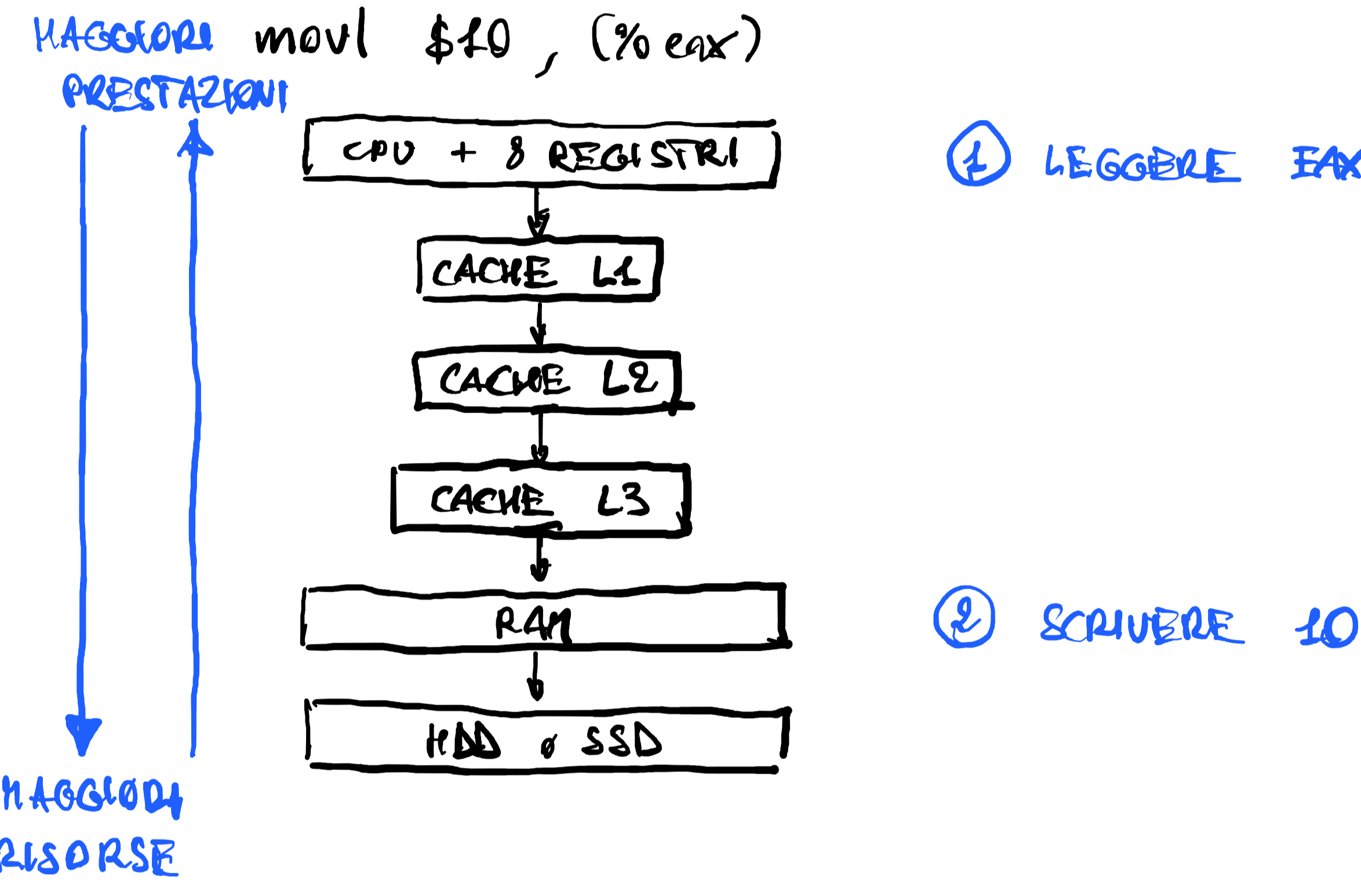
SPAZIO : memoria utilizzata

ENERGIA : consumo energetico

THROUGHPUT : numero di operazioni completate per unità di tempo

ANALISI PRESTAZIONALE

Quanto tempo richiede (latenza) l'esecuzione di



Evento ²⁰	Latenza effettiva	Latenza riscalata
Ciclo di clock	0.4 nsec	1 sec
Accesso cache L1	0.9 nsec	2 sec
Accesso cache L2	2.8 nsec	7 sec
Accesso cache L3	28 nsec	1 min
Accesso RAM DDR3 DIMM	100 nsec	4 min
Accesso disco SSD	50-150 μsec	1.5-4 giorni
Accesso disco a rotazione	1-10 msec	1-9 mesi
Invio pacchetto Internet continentale/intercontinentale	65-141 msec	5-11 anni

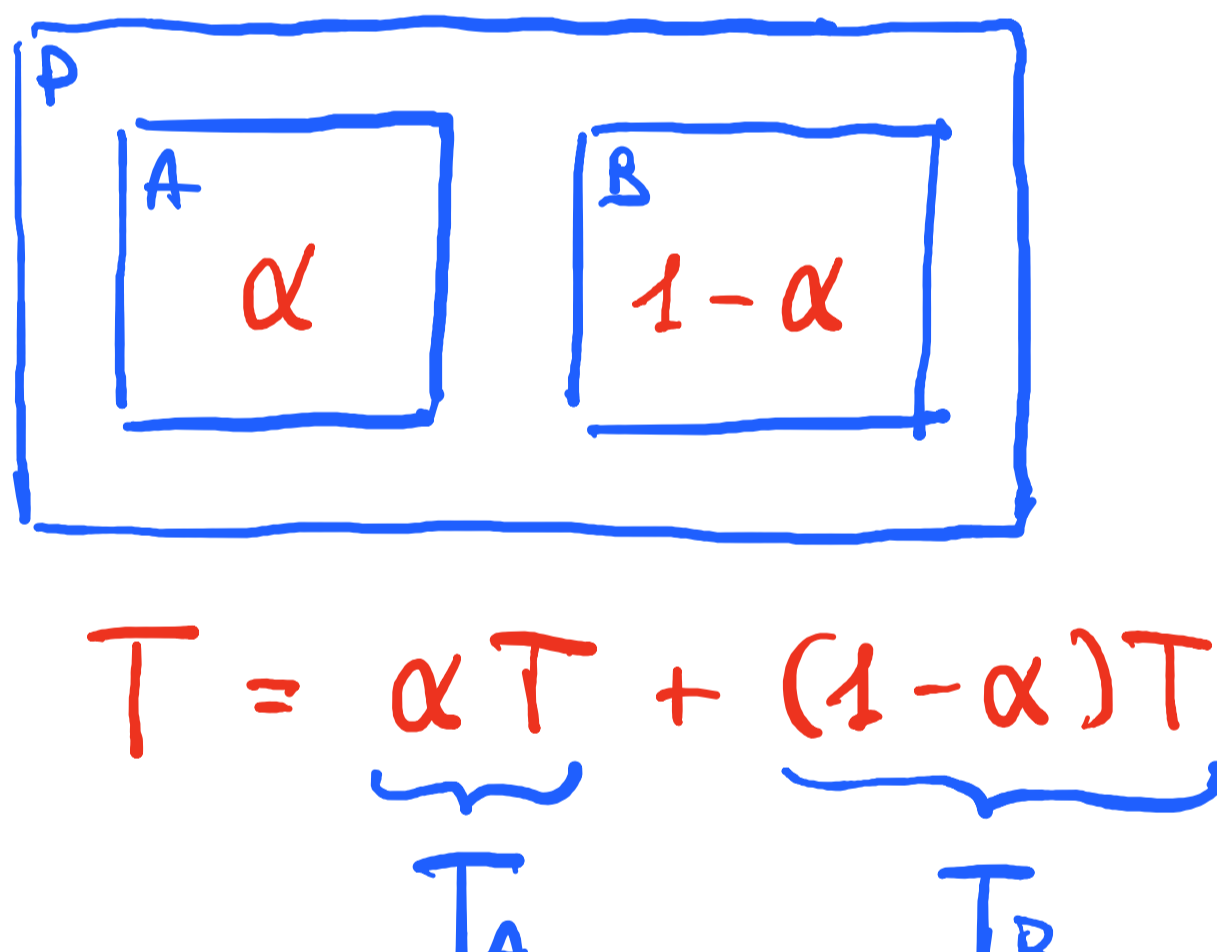
Come misuro il miglioramento prestazionale ?

$$SPEEDUP_0 = \frac{T_0}{T'_0}$$

T_0 ← latenza del programma non ottimizzato (senza O)
 T'_0 ← latenza del programma ottimizzato con O
 Esempio: $2,051 \approx 2x$

LEGGE DI AMPDALL

Ci permette di calcolare lo speedup complessivo di un programma a fronte di ottimizzazione di una sua parte



IPOTESI : posso ottimizzare A con uno speedup V

Qual'è lo speedup di P ?

$$S = \frac{T}{T'} \quad T' = T'_A + T_B = \frac{T_A}{k} + T_B = \frac{\alpha T}{k} + (1-\alpha) T = \left(\frac{\alpha}{k} + 1 - \alpha \right) T$$

LEGGE DI AMPDALL

$$S = \frac{T}{T'} = \frac{T}{\left(\frac{\alpha}{k} + 1 - \alpha \right) T} = \frac{1}{\frac{\alpha}{k} + 1 - \alpha}$$

← frazione di tempo spesa per eseguire A
 ← speedup di A

Es.: $\alpha = 0,5 \quad k=2 \rightarrow S = 1,33x$

Es.: $\alpha = 0,9 \quad k=6 \rightarrow S = 6x$

Qual'è lo speedup teorico massimo ?

$$\lim_{k \rightarrow \infty} \frac{1}{\frac{\alpha}{k} + 1 - \alpha} = \frac{1}{1 - \alpha}$$

PROFILAZIONE DELLE PRESTAZIONI

Calcolare il tempo speso dal programma nell'esecuzione di ciascuna delle sue parti

MANUALE : tramite strumentazione di codice

AUTOMATICA : uso di Tool

PROFILAZIONE MANUALE

```

1) long start = misura tempo(); // in alternativa, usare double
2) // porzione di codice da misurare
3) // ...
long elapsed = misura tempo(); - start;
printf("Tempo richiesto: %ld\n", elapsed);
    
```

Arricchimento al TEMPO STANDARD UNIX

1/1/1970 00:00:00

```

long get_real_time_msec() {
    struct timespec ts;
    clock_gettime(CLOCK_MONOTONIC, &ts);
    return ts.tv_sec*1000 + ts.tv_nsec/1000000;
}

long get_user_time_msec() {
    struct rusage ru;
    getrusage(RUSAGE_SELF, &ru);
    return ru.ru_utime.tv_sec*1000 + ru.ru_utime.tv_usec/1000;
}

double get_sys_time_msec() {
    struct rusage ru;
    getrusage(RUSAGE_SELF, &ru);
    return ru.ru_stime.tv_sec*1000 + ru.ru_stime.tv_usec/1000;
}
    
```

