



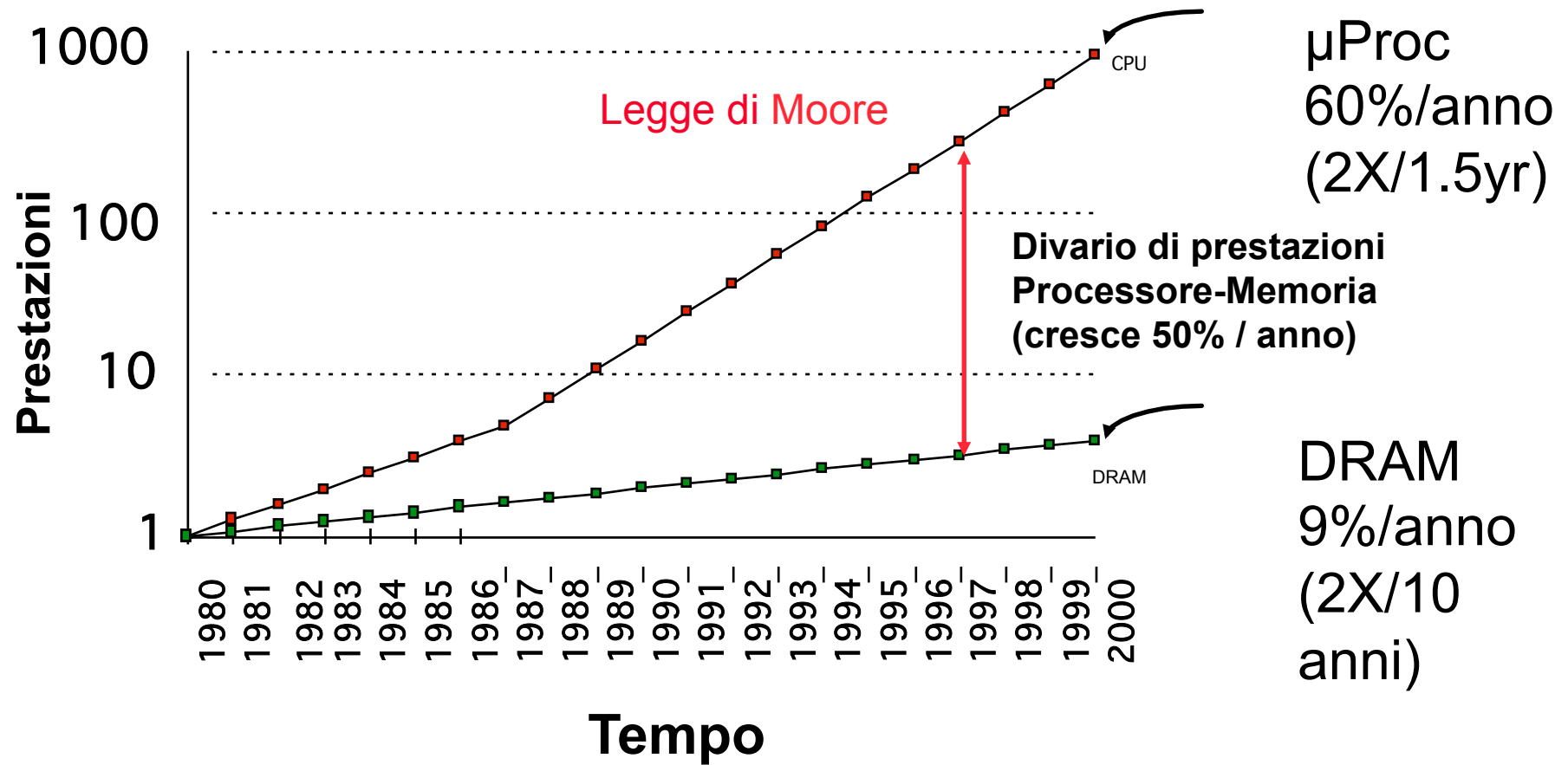
La Memoria Cache

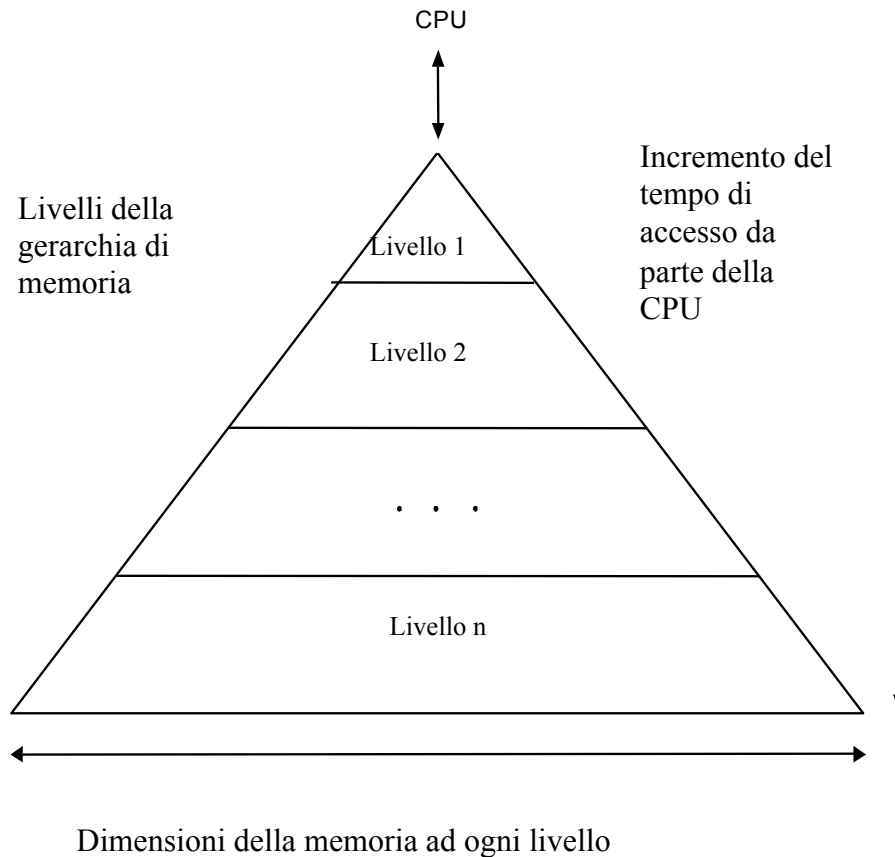
Informatica B

Il problema della memoria

- ❑ Obiettivo:
 - ▶ fornire agli utenti una memoria grande e veloce
 - ▶ fornire al processore i dati alla velocità con cui è in grado di elaborarli
- ❑ Problema: Il tasso di crescita nella velocità dei processori non è stato seguito da quello delle memorie
 - Tempo di accesso alle SRAM: 2 - 25ns al costo di \$100 - \$250 per Mbyte.
 - Tempo di accesso alle DRAM: 60-120ns al costo di \$5 - \$10 per Mbyte.
 - Tempo di accesso al disco: da 10 a 20 million ns al costo di \$.10 - \$.20 per Mbyte.

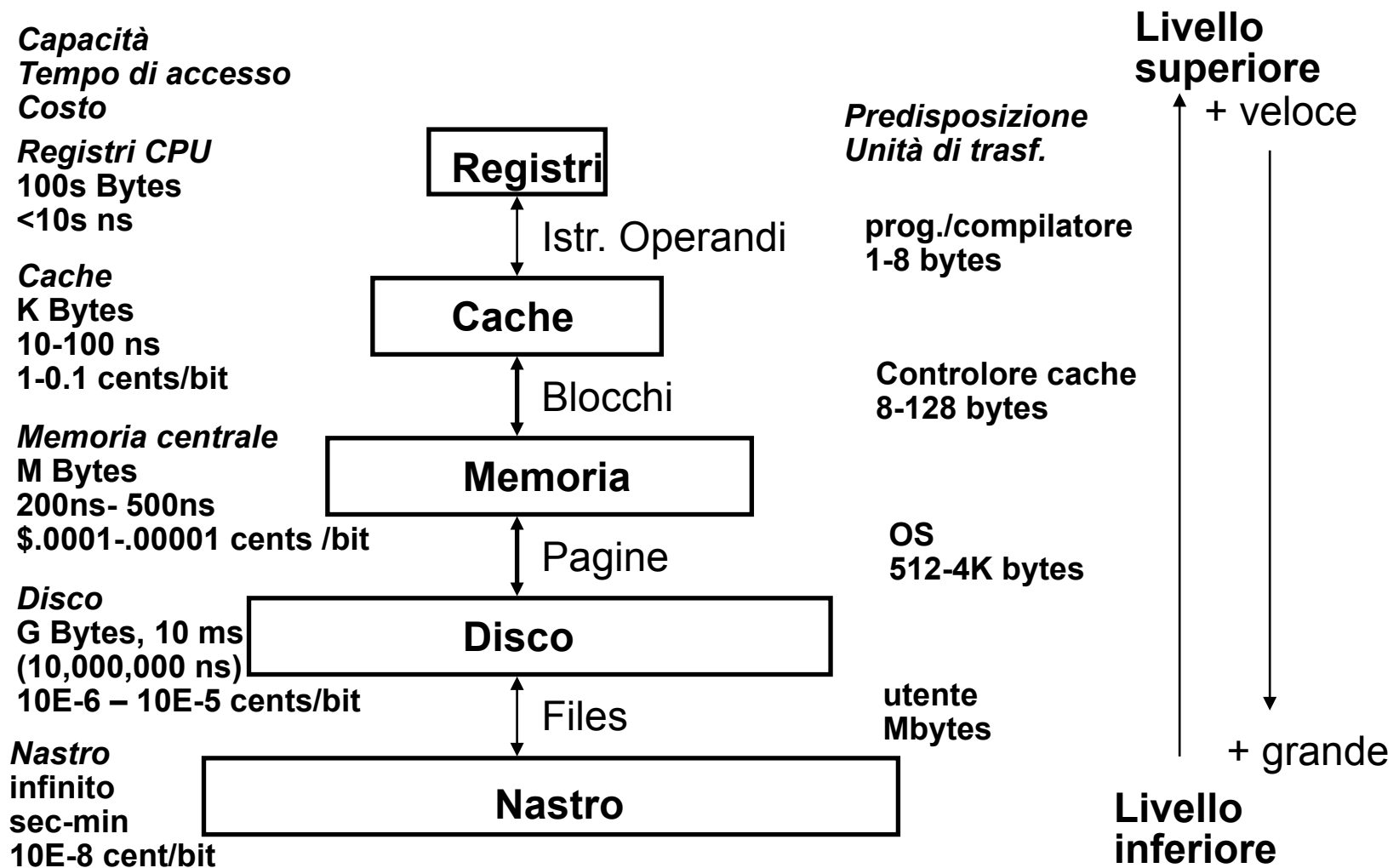
Prestazioni di processori e cache





Utilizzare diversi livelli di memoria, con tecnologie diverse in modo da ottenere un buon compromesso costo/prestazioni

Livelli della gerarchia di memoria



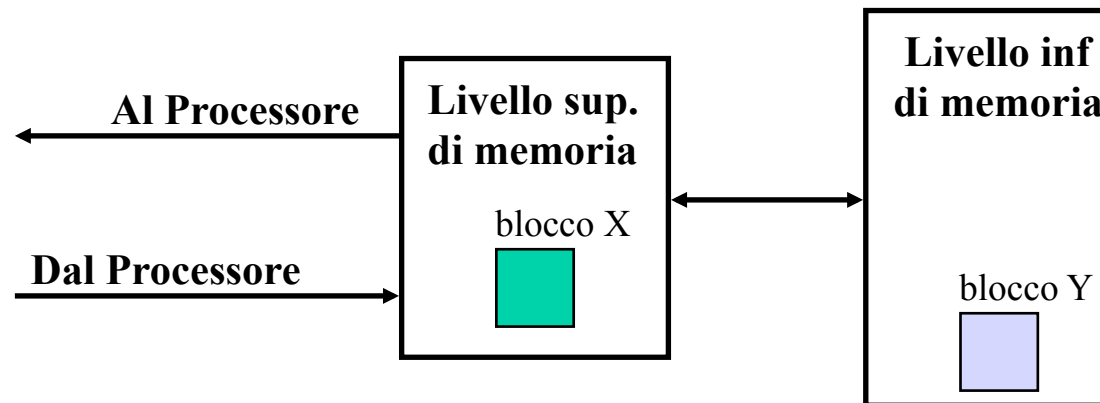
Località

- ❑ Il principio che rende la gerarchia di memoria una buona idea per incrementare le prestazioni del sistema di memoria
- ❑ Località: in ogni istante di tempo un programma accede a una parte relativamente piccola del suo spazio di indirizzamento
- ❑ Esistono due diversi tipi di località: *temporale* e *spaziale*

Il principio di località

- ❑ **Località temporale:** se un dato viene referenziato in un dato istante, è probabile che lo stesso dato venga nuovamente richiesto entro breve
- ❑ **Località Spaziale:** Se un dato viene utilizzato in un dato istante, è probabile che dati posizionati in celle di memoria adiacenti vengano anch'essi richiesti entro breve
- ❑ Negli ultimi 15 anni, le tecniche di miglioramento delle prestazioni nell'hardware si sono basate sul principio di località

- ❑ Si considerino solo due livelli di gerarchia:
- ❑ Il processore richiede un dato al sistema di memoria:
 - ▶ La richiesta viene prima inviata al livello di memoria superiore (più vicino al processore)
 - ▶ Se il dato non è presente nel livello superiore (fallimento della richiesta) la ricerca viene effettuata nel livello inferiore



Gerarchia di memoria: definizioni

- ❑ **Hit (successo)**: dati presenti in un blocco del livello superiore (esempio: Blocco X)
 - ▶ **Hit Rate (tasso di successo)**: numero di accessi a memoria che trovano il dato nel livello superiore sul numero totale di accessi
 - ▶ **Hit Time (tempo di successo)**: tempo per accedere al dato nel livello superiore della gerarchia:
Tempo di accesso alla cache + tempo per determinare successo/fallimento della richiesta

Gerarchia di memoria: definizioni

- ❑ **Miss (fallimento)**: i dati devono essere recuperati dal livello inferiore della memoria (Blocco Y)
 - ▶ **Miss Rate (tasso di fallimento)** = $1 - (\text{Hit Rate})$
 - ▶ **Miss Penalty (tempo di fallimento)**: tempo necessario a sosituire un blocco nel livello superiore + tempo per trasferire il blocco al processore
 - ▶ **Hit Time** \ll **Miss Penalty**

Cache

- ❑ Memoria al livello superiore della gerarchia
- ❑ Sfruttare il principio di località dei programmi e tenere in memoria cache i dati utilizzati più di recente
- ❑ Obiettivo: fornire dati al processore in uno o due cicli di clock
- ❑ Memoria cache: veloce nei tempi di accesso ma di dimensioni ridotte

- ❑ Considerate i seguenti sistemi di memoria:
 - ▶ Sistema A: memoria centrale con memoria cache che ha le seguenti caratteristiche:
 - Hit Rate = 70%
 - Hit Time = 10ns
 - Miss Penalty = 300ns
 - ▶ Sistema B: Una memoria centrale senza memoria cache con un tempo medio di accesso di 90ns

- ❑ Rispondere alle seguenti domande:

- ▶ Quale dei due sistemi di memoria è migliore?

Calcoliamo il tempo medio di accesso alla memoria a) T:

$$T = HR*HT+(1-HR)*MP = 0.70*10ns+0.30*300ns = 97ns$$

→ Il sistema B è migliore

- ❑ Considerate i seguenti sistemi di memoria:
 - ▶ Una memoria centrale con memoria cache che ha le seguenti caratteristiche:
 - Hit Rate = 70%
 - Hit Time = 10ns
 - Miss Penalty = 300ns
 - ▶ Una memoria centrale senza memoria cache con un tempo medio di accesso di 90ns
- ❑ Rispondere alle seguenti domande:
 - ▶ Cambierebbe la vostra risposta se l'Hit Rate fosse pari all'80% ?

Se $HR=80\%$ il tempo medio di accesso diventa

$$T' = HR' * HT + (1 - HR') * MP = 0.80 * 10ns + 0.20 * 300ns = 68ns$$

→ Il sistema A diventerebbe migliore