04. Gerarchie di Memoria e Cache

Caratteristiche delle Memorie

Le memorie si classificano secondo:

- **Locazione**: processore, principale (interna), secondaria (esterna)
- **Capacità**: dimensione parola, numero di parole
- **Unità di trasferimento**: parola, blocco
- **Metodo di accesso**: sequenziale, diretto, casuale, associativo
- **Prestazioni**: tempo di accesso, tempo di ciclo, velocità di trasferimento
- **Modello fisico**: semiconduttore, magnetico, ottico, magneto-ottico
- **Caratteristiche fisiche**: volatile/non volatile, riscrivibile/non riscrivibile

La memoria ideale sarebbe ampia, veloce ed economica. Le tecnologie esistenti impongono un compromesso:

- **Registri e cache**: veloci, costosi, piccoli
- **SRAM/DRAM**: compromesso intermedio
- **Dischi, CD/DVD, nastri**: lenti, economici, capienti

Principio della Gerarchia di Memoria

Le CPU hanno migliorato le prestazioni grazie a innovazioni tecnologiche e architetturali; le memorie solo grazie ad avanzamenti tecnologici, creando un divario prestazionale crescente.

Proprietà dei Programmi

I programmi esibiscono **località dei riferimenti**:

- **Linearità**: indirizzi acceduti spesso consecutivi
- **Località spaziale**: accessi ad indirizzi contigui più probabili
- **Località temporale**: la zona acceduta di recente è quella più probabile per accessi futuri

La **congettura 90/10** afferma che un programma impiega il 90% del tempo di esecuzione sul 10% delle istruzioni.

Organizzazione Gerarchica

- Livelli inferiori: supporti più capaci, lenti e meno costosi
- Livelli superiori: supporti più veloci, costosi e meno capaci
- La CPU utilizza direttamente il livello più alto (cache)
- Ogni livello inferiore contiene tutti i dati presenti ai livelli superiori
- Trasferimento CPU-cache: per parola
- Trasferimento cache-memoria centrale: per blocco

Suddivisione in Blocchi

La memoria è suddivisa in **blocchi**, unità minima indivisibile di trasferimento dal livello inferiore. L'indirizzo di un dato è composto da:

- Indirizzo del blocco
- Posizione del dato all'interno del blocco

Hit e Miss

- **Hit**: dato richiesto presente in cache
- **Miss**: dato assente in cache

L'hit deve essere molto probabile (>90%) per ottenere efficienza. Un miss avvia una procedura di **swap** con il livello inferiore.

Tempo Medio di Accesso

$$T_a = T_h \cdot P_h + T_m \cdot (1 - P_h)$$

- T_a: tempo medio di accesso di un dato in memoria
- T_h: tempo di accesso in caso di hit
- T_m: tempo di accesso in caso di miss (funzione della dimensione del blocco)
- P_h: probabilità di hit (funzione della dimensione del blocco e della politica di gestione)

Tecnica Generale di Funzionamento

1. La memoria centrale è suddivisa in blocchi logici
2. La cache è dimensionata come multiplo di blocchi
3. Per ogni indirizzo emesso dalla CPU:
   - **Hit**: il dato è fornito immediatamente
   - **Miss**: la cache richiede il dato al livello inferiore → il blocco viene caricato in cache → il dato viene fornito alla CPU

La cache può essere:

- **Logica**: opera su indirizzi logici, prima della MMU
- **Fisica**: opera su indirizzi fisici, dopo la MMU

Tecniche di Associazione

Associazione Diretta (Direct Mapping)

Ogni blocco del livello inferiore può essere allocato in **una sola specifica posizione** (linea/slot) del livello superiore.

Formula: ILS = ILI mod N

- ILS = Indirizzo Livello Superiore
- ILI = Indirizzo Livello Inferiore
- N = Numero di blocchi

Struttura indirizzo: etichetta (tag) | linea | parola

Vantaggi:

- Semplicità di traduzione
- Determinazione veloce di hit/miss

Svantaggi:

- Necessità di etichetta (tag) per identificare il blocco presente
- Swap frequenti per accesso a dati di blocchi adiacenti che mappano sulla stessa linea

Associazione Completa (Fully Associative)

Ogni blocco del livello inferiore può essere posto in **qualunque posizione** del livello superiore.

Struttura indirizzo: etichetta (tag) | parola

Vantaggi:

- Massima efficienza di allocazione

Svantaggi:

- Determinazione onerosa della corrispondenza ILS-ILI
- Verifica hit/miss richiede confronto con tutte le etichette

Associazione a Gruppi (K-way Set Associative)

Ogni blocco di un insieme del livello inferiore può essere allocato **liberamente in uno specifico gruppo** di K posizioni del livello superiore.

Struttura indirizzo: etichetta (tag) | set | parola

Valutazione: compromesso ottimale tra associazione diretta e completa, con buona efficienza di allocazione e complessità di ricerca accettabile. L'hit ratio migliora all'aumentare del grado di associatività e della dimensione della cache.

Politiche di Rimpiazzo dei Blocchi

Determinano quale blocco sostituire in cache durante uno swap:

- **Casuale**: occupazione omogenea dello spazio
- **FIFO**: sostituisce il blocco rimasto più a lungo in cache
- **LFU (Least Frequently Used)**: sostituisce il blocco con meno accessi
- **LRU (Least Recently Used)**: preserva la località temporale

LRU offre probabilità di miss inferiori rispetto al rimpiazzo casuale, specialmente con cache piccole e alta associatività.

Il Problema della Scrittura

La scrittura genera incoerenza tra blocco in cache e blocco nei livelli inferiori.

Write Through

- Scrittura contemporanea in cache e nel livello inferiore
- Dati sempre coerenti tra i livelli
- Aumento del traffico per scritture frequenti nello stesso blocco
- Si utilizzano buffer di scrittura asincroni

Write Back

- Scrittura in memoria inferiore differita al rimpiazzo del blocco
- Richiede un dirty bit per ricordare se sono avvenute scritture
- Ottimizza il traffico tra livelli
- Causa periodi di incoerenza (problematico con I/O e multiprocessori)

Coerenza nei Sistemi Multiprocessore

Con più processori con cache locale e memoria condivisa, la modifica in una cache invalida la parola corrispondente in memoria centrale e nelle altre cache.

Soluzioni:

- **Monitoraggio del bus**: controllori cache intercettano modifiche a locazioni condivise
- **Trasparenza hardware**: hardware aggiuntivo propaga modifiche a tutte le cache
- **Memoria non-cacheable**: porzione condivisa di memoria non viene cachata

---

Tipi di Miss

- **Miss di primo accesso**: inevitabile e non riducibile
- **Miss per capacità insufficiente**: la cache non può contenere tutti i blocchi necessari
- **Miss per conflitto**: più blocchi mappano sulla stessa linea/gruppo

---

Tecniche di Riduzione dei Miss

Tecniche Classiche

- **Maggiore dimensione di blocco**: sfrutta località spaziale, ma aumenta miss per conflitto
- **Maggiore associatività**: aumenta tempo di localizzazione; vale la **regola del 2:1** (cache a N blocchi con associazione diretta ≈ cache N/2 con associazione a 2 vie)

Altre Tecniche

- Cache multilivello (L1, L2, L3)
- Separazione tra cache dati e cache istruzioni
- Ottimizzazione mediante compilatori:
  - Posizionamento accurato di procedure ripetitive
  - **Fusione di vettori in strutture**: trasforma array separati in array di strutture per migliorare località spaziale
  - **Trasformazione di iterazioni annidate**: scambia ordine dei cicli per accedere alla memoria in modo sequenziale