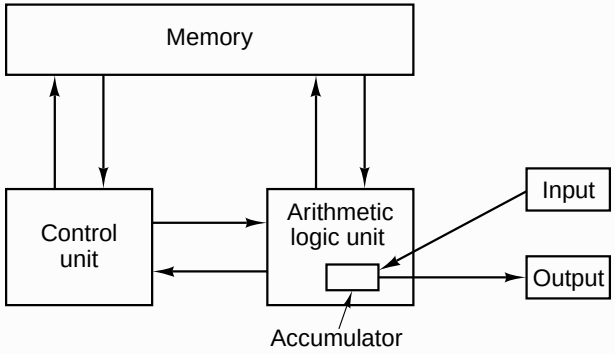
Architetture parallele

Modello di calcolo: macchina di von Neumann.



Macchina di von Neumann

Limite alle prestazione ottenibili:

- un'unica operazione in esecuzione
- una sola parte attiva

Ricerca di nuove architetture con più operazioni in esecuzione allo stesso istante.

Cenni storici

Molte proposte per architetture parallele:

- Reti neuronali,
- Date flow machine,
- Calcolatori vettoriali,
- Connection machine,
- Illiac IV.

Scarso successo commerciale.
I calcolatori vettoriali usate per supercomputer per il calcolo scientifico.
Approcci rivoluzionari, difficili da usare,
richiedono nuovi stili di programmazione.

Cenni storici

Macchine parallele a larga diffusione: anni 90.

- Pipeline, superscalari
  Approcci conservativi, mantengo il codice.
- processore multi-core, sistemi multiprocessore, cluster di calcolatori.
  Ultime evoluzioni, conseguenze della:
  legge di Moore,
  vincoli sui consumi energetici.

  Per sfruttare a piene il parallelismo devo usare algoritmi concorrenti distribuiti.

Illustreremo

- problematiche e aspetti generali del parallelismo;
- classificazione dei calcolatori paralleli;
- esempi di calcolatori paralleli, idee architetturali attualmente in uso.

Motivazioni per il parallelismo

Migliorare le prestazioni:

- problemi che richiedo molta potenza computazionale: algoritmi di simulazione;
- il software tende a sfruttare tutte le risorse disponibili.

Aumento delle prestazioni mediante:

- nuove tecnologie (porte logiche più veloci).
- nuove architetture (computazione più efficiente, maggior parallelismo)

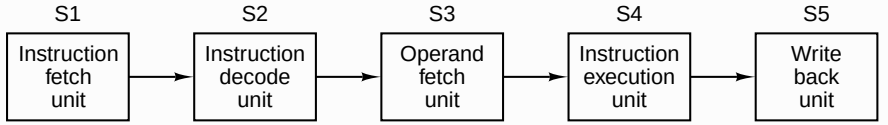
Motivazioni per il parallelismo

Altri vantaggi:

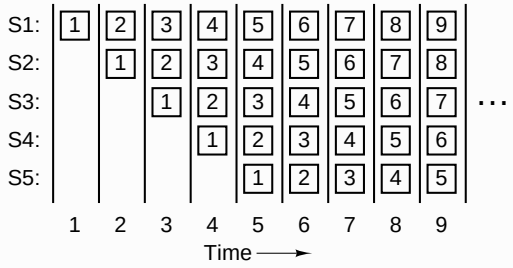
- tolleranza ai guasti: più unita di calcolo, un guasto non necessariamente blocca il calcolatore,
- architettura scalabile: posso aumentare le prestazioni aggiungendo nuove unità di calcolo.

Parallelismo su un singolo chip

Pipeline: divisione della (micro-)istruzione in stadi.

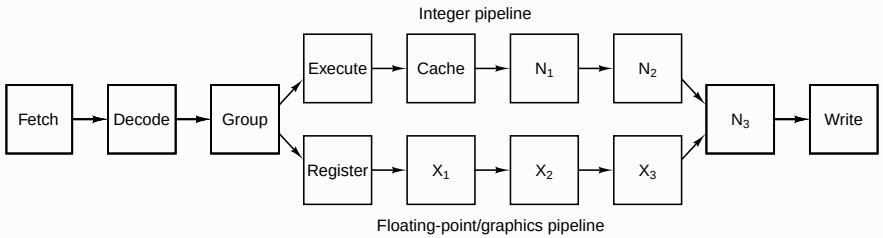


(a)



(b)

Processori superscalari



Più istruzioni iniziate contemporaneamente

Prestazioni: aumento di un fattore circa dieci

Parallelismo a livello di istruzione

Più istruzioni, di uno stesso processo, eseguite contemporaneamente.

- Vantaggi: si utilizza codice sequenziale, facile da utilizzare, non è necessario riscrivere il codice.
- Svantaggi: difficile ottenere un elevato grado di parallelismo.

Parallelismo su di un singolo chip

Altre tecniche il parallelismo:

- Processori VLIW Very Long Instruction Words
- On-Chip Multi-threading
- Single-Chip Multiprocessor
  - Multiprocessori omogenei
  - Multiprocessori eterogeneo

VLIW – Very Long Instruction Words

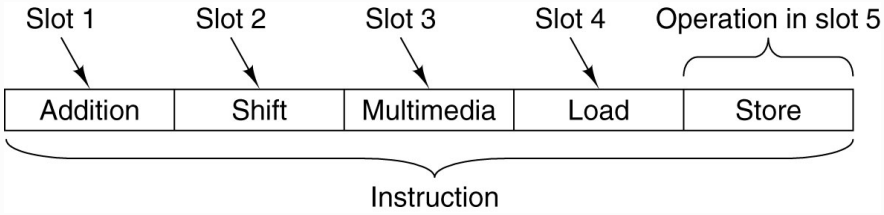
Nuovi linguaggi macchina adatti alla computazione parallela.
Sfruttano al meglio le tecnologie esistenti.

I nuovi linguaggi macchina usano:
VLIW – Very Long Instruction Words ossia istruzioni molto lunghe, ciascuna composta da una sequenza di istruzioni base.

Istruzione base da eseguire in parallelo contemporaneamente.
Senza controllare che siano indipendenti tra loro.
Sposto il lavoro dal processore al compilatore.

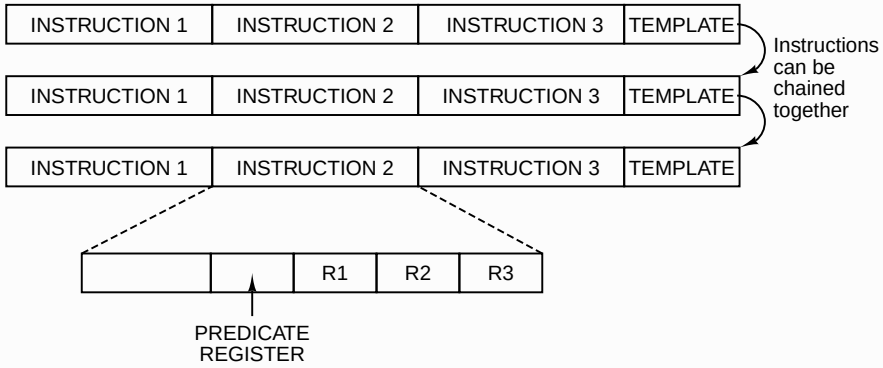
TriMedia

Processore sviluppato dalla Philips, per dispositivi embedded di audio-video (DVD player - recorder, camcorder, ...
Una singola istruzione è composta da 5-8 sotto-istruzioni: operazioni aritmetiche, load-store, multimediali (vettoriali).



Itanium IA-64

Intel, candidato a sostituire l'IA-32 (Pentium, Core).



Più istruzioni possono essere collegate assieme ed eseguiti contemporaneamente,
senza controllare l'indipendenza.

Itanium IA-64

- Istruzioni condizionate, per evitare le istruzioni di salto.
- Centinaia di registri, evito:
  accessi in memoria,
  dipendenza tra istruzioni.
- Caricamenti speculativi,
  anticipo gli accessi in memoria.
  a meno di page-fault.

Poco successo commerciale,
il mercato ha preferito x86-64 (o AMD 64)
scelta conservativa.

Multi-threading

Permette di sfruttare meglio le capacità di calcolo dei processori superscalari (con più pipeline).

Il processore esegue più thread (processi (con memoria condivisa)) contemporaneamente.

Utile nel caso un programma rallenti per:

- dipendenze tra istruzioni,
- istruzioni che bloccano, per alcuni cicli di clock l'esecuzione (accessi alla memoria principale, cache III livello)

Multi-threading

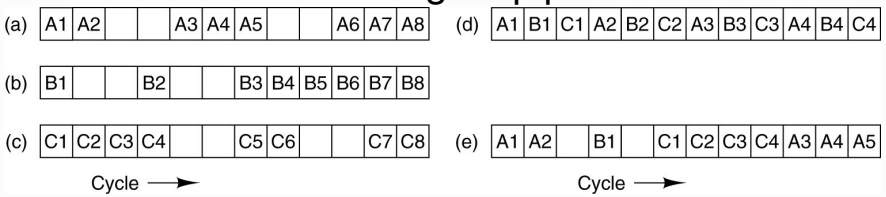
Può essere utilizzato in processori con singola pipeline.
Più utile in processore superscalari.

In determinati istanti il processore commuta da un thread ad un altro.
Due tecniche possibili;

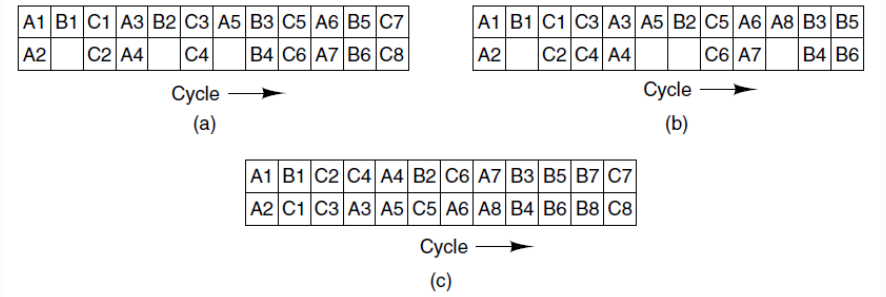
- m. grana fine: si commuta su ogni, numero elevato di commutazione di contesti, si anticipano i possibili blocchi.
- m. grana grossa: si cerca di eseguire più istruzioni per thread, riduco le commutazioni.

Multi-threading

Processore con una singola pipeline:



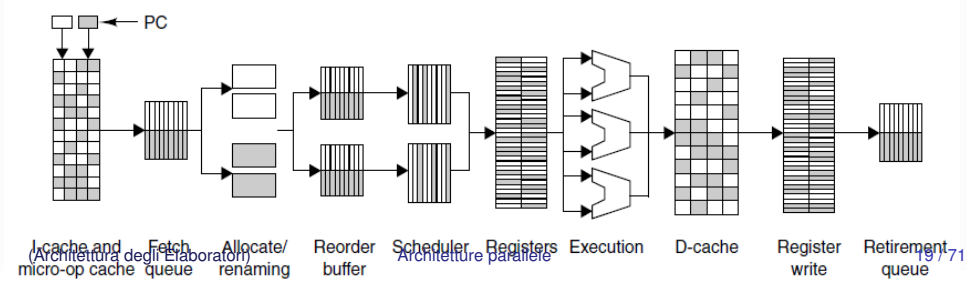
Processore superscalare:



Ripartizione risorse

Come le risorse del processore vengono ripartite tra i vari thread.
Esempi:

- ogni thread disponi di un suo insieme di registri:
- risorse condivise: memoria cache
- risorse partizionate: le pipeline di un processore super-scalare



Ripartizione risorse

In generale:

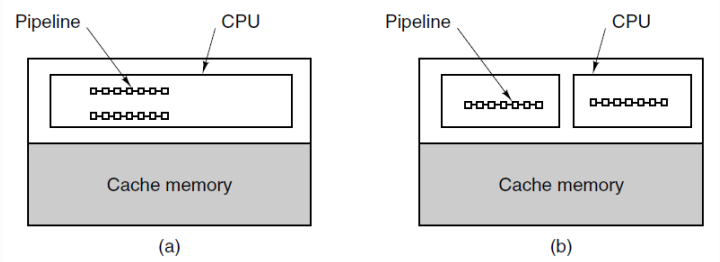
- condivisione ripartita: ogni thread un insieme privato di risorse,
- condivisione totale,
- condivisione a soglia; un limite alle risorse acquisibili da un thread.

Multi-threading usato su molti processori:

- Core i7: hyper-threading, 2 thread/core,
- UltraSparcT3: 16 core, 8 thread/core

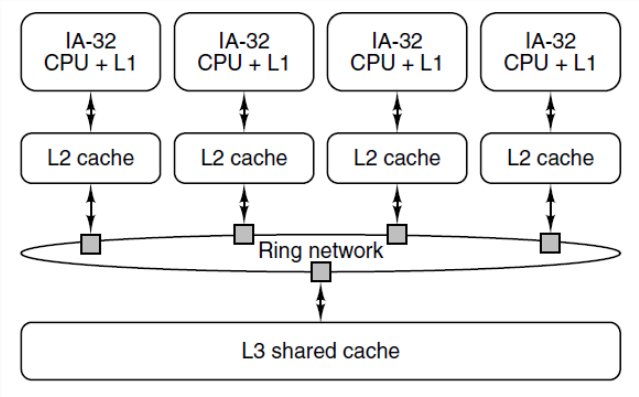
Multiprocessori omogenei su un singolo chip

Evoluzione del multi-threading, ogni thread un core.
Totale separazione delle risorse tra i thread.
Solo a memoria cache resta comune.



La tecnologia attuale permette di inserire più processori su di un chip.

Intel Core i7



Maggior numero di core:
- UltraSparc
- Celle (Sony, Toshiba, IBM)

Co-processori

Parallelismo ottenuto delegando alcuni compiti a processori ausiliari.

- DMA (Direct Memory Access): controllore DMA complesso, co-processore, si fa carico della gestione I/O.
- Scheda video: generazione di immagini.
- Co-processori multimediali: la gestione dei dati audio video gravosa: codifica - decodifica (MPEG, MP3), elaborazione del segnale.
- Scheda di rete: controllo degli errori, instradamento dei messaggi.
- Cripto-processori.

Multipr. eterogenei su singolo chip

In un chip:

- core principale di controllo
- co-processori specializzati su particolari applicazioni
- bus di interconnessione, diversi standard alternativi:
  CoreConnect IBM, AMBA (Advance Microcontroller Bus Architecture) CPU ARM, VCI (Virtual Component Interconncect)

Chip progettato combinando i progetti delle singole componenti (core).
Dispositivi embedded: SoC (System on Chip),

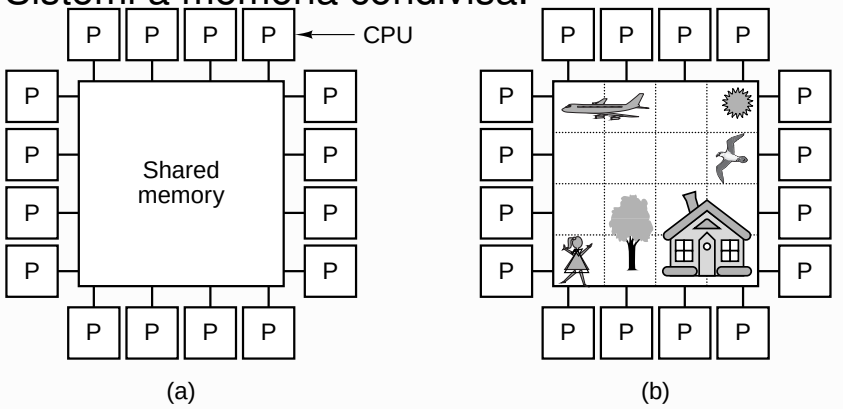
Parallelismo su più chip

Più processori su più circuiti integrati.

- Maggiori aumenti di prestazioni.
- Necessità di nuovo codice (o di un adatto SO), nuove problematiche.
- Considereremo due principali approcci: multiprocessori e multicomputer.

Multiprocessori

Sistemi a memoria condivisa.

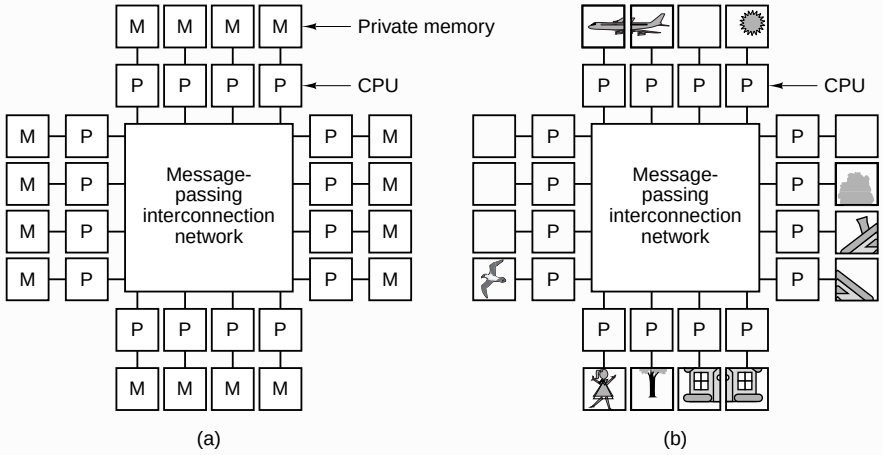


Multiprocessori

- Le CPU dividono lo stesso spazio di memoria.
- Comunicazione tra CPU attraverso l'accesso alla memoria condivisa LOAD, STORE.
- Non è necessario ripartire i dati tra le CPU
- Accesso alla memoria limita le unità di computazione (il grado di parallelismo).
- Esempi: PC con multi-processore, server.

Multicomputer

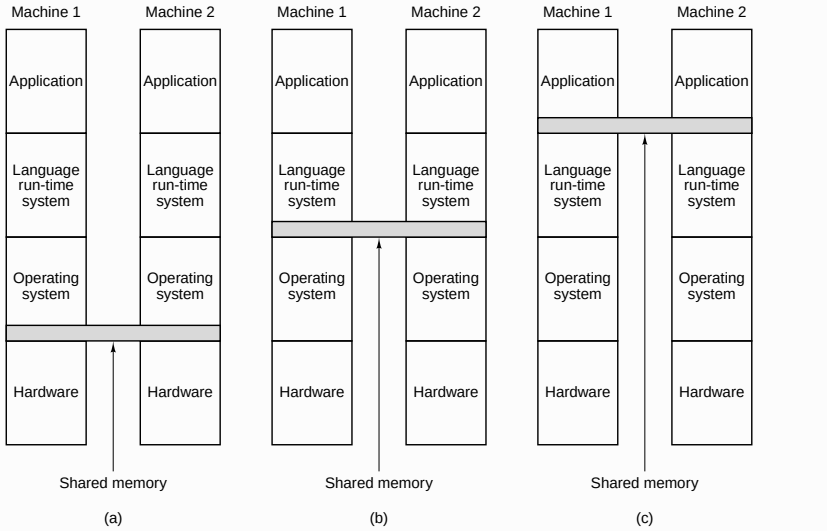
Sistemi a memoria distribuita:



Multicomputer

- Ogni CPU ha un proprio spazio di memoria.
- Comunicazione attraverso scambio di messaggi: SEND, RECEIVE.
- Più semplice integrare numerose CPU, numero unità limitato dalla capacità della rete di interconnessione.
- Più difficili da programmare.

Simulazione memoria condivisa



Simulazione memoria condivisa

1. Memoria condivisa hardware.
2. Memoria condivisa tramite memoria virtuale, paginazione.
   Una pagina può trovarsi:
   - nel memoria della CPU,
   - in memoria disco,
   - nella memoria locale di un altra CPU.

   DSM (Distributed Shared Memory).
3. Memoria condivisa tramite librerie.
   - Linda: programmi con tuple condivise.
   - Orca: programmi con oggetti condivisi.

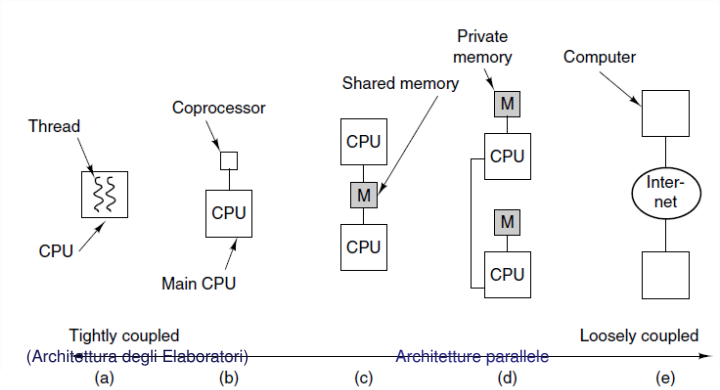
Catalogazione del parallelismo

Granularità: complessità delle computazioni eseguite in parallelo (e dei circuiti che le eseguono)

- Fine: semplici istruzioni, Instruction Level Parallelism (ILP). Es. processori con pipeline.
- Grossa (coarse): procedure, Process Level Parallelism (PLP). Es. processori multi-threading, sistemi multiprocessori.

Livello di accoppiamento

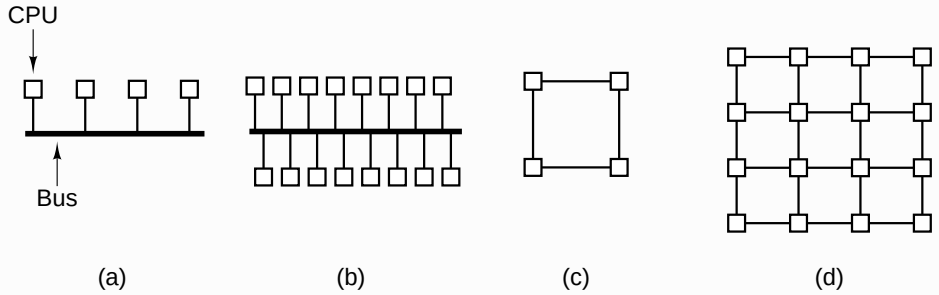
- Forte: unità fortemente connesse, notevole scambio di dati. Es. processori superscalari, multiprocessori.
- Debole: unità più indipendenti.
  Es. multicomputer.



Scalabilità

Una stessa idea architetturale può essere implementata con un numero variabile di unità di calcolo.

Un'architettura è scalabile se funziona correttamente con molte unita di calcolo.



Miglioramento delle prestazioni

Problemi:

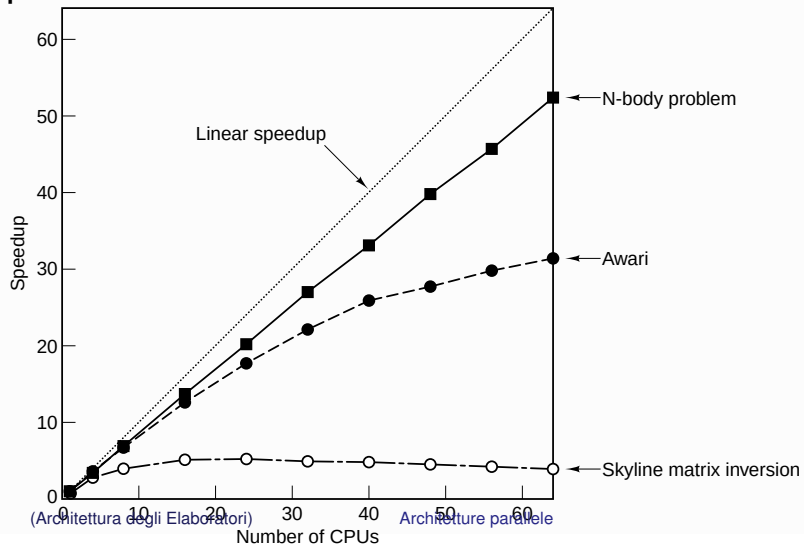
- Le unità di calcolo devono sincronizzarsi tra di loro, overhead di computazione (lavoro extra).
- Memoria, o altre risorse condivise, creano conflitti tra le unità di calcolo.

Conseguenze:

- difficilmente *n* unità svolgono un lavoro *n* volte più velocemente,
- a volte un aumento delle unità di calcolo porta a un peggioramento delle prestazioni.

Prestazione

Scalabilità dipende sia dall'architettura che dai problemi considerati:



Componenti di un calcolatore parallelo

- Unità di calcolo. Si usano processori standard.
- Unità di memoria. Spesso partizionata per poter essere condivisa tra più processori.
- Rete di interconnessione.
  - Nei multiprocessori collegano processori e moduli di memoria.
  - Nei multicomputer collegano le unità di calcolo.

Memoria

- Composta da diverse unità operanti in parallelo.
- Struttura complessa, dati replicati:
  - ogni core possiede una cache locale,
  - cache di alto livello comune a più core,
  - memoria principale comune a più processori, eventualmente divisa su più unità.
- Tempi di accesso ai dati dipendo dalla loro posizione rispetto al processore.
  Per preservare l'efficienza, sono ammessi accessi alla memoria fuori ordine:
  i dati disponibili vengono subito letti o scritti.

Consistenza della memoria

Una memoria con comportamento ideale (esecuzione in ordine, tutti i processore hanno la stessa visione della memoria) è troppo lenta.
Ma l'ordine di accesso ai dati in memoria non può essere arbitrario,
vanno stabilite regole minime per rendere prevedibile il comportamento del codice.

- Sequenziale: tutte le CPU vedono lo stesso ordine.
- Di processore: preserva: le scritture di ogni CPU, la scrittura di ogni indirizzo.
- Debole: prevede richieste di sincronizzazione,
- Dopo il rilascio: sincronizzazione locale.

Reti di interconnessione

Equivalente sofisticato dei bus.
Componente fondamentale: a volte la componente più costosa del calcolatore.

Composte da:

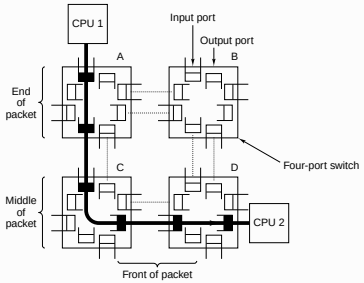
- Link (connessioni - cavi - bus)
  paralleli o seriali
- Switch (instradatori - commutatori)
- Interfacce (Processori - rete - memorie)

Switch

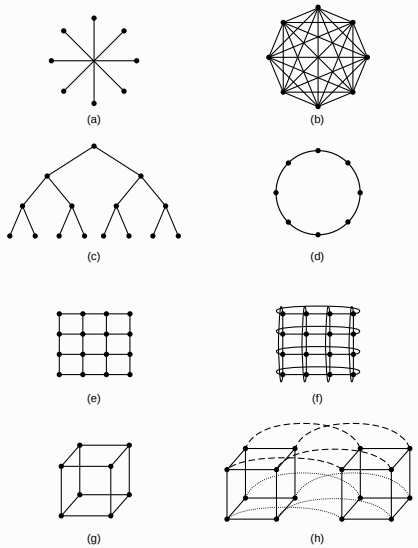
Metodi di comunicazione:

- Circuit switching,
- Store and forward: messaggio diviso in pacchetti.
  input, output, common buffering

Routing: determinazione del percorso.



Topologie di rete



Topologie di rete

Modo di strutturare le unità di calcolo:

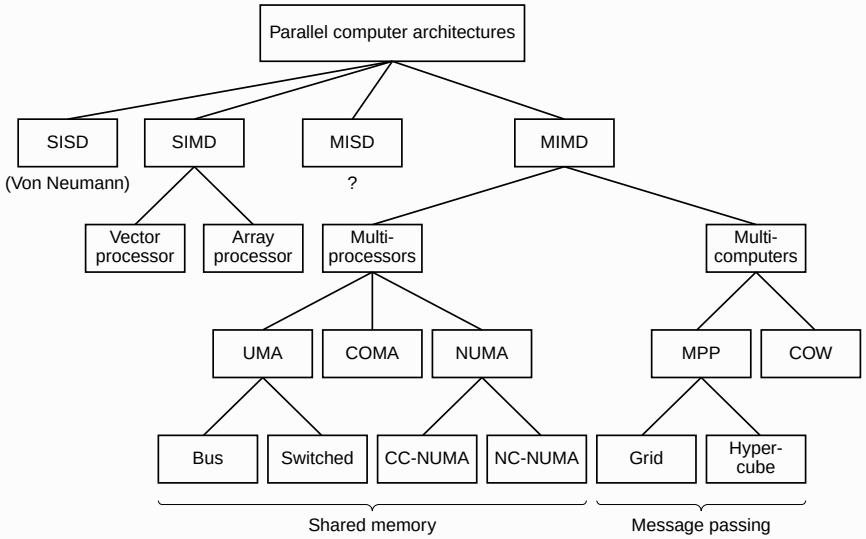
- Obiettivi
  - massimizzare la ampiezza di banda,
  - evitare colli di bottiglia,
  - minimizzare le distanze (in numero di archi): diametro.
- Configurazione: stella, completamente interconnessa, albero, griglia, doppio toroide, ipercubo.
  geometria fissa o variabile

Tassonomia dei computer paralleli

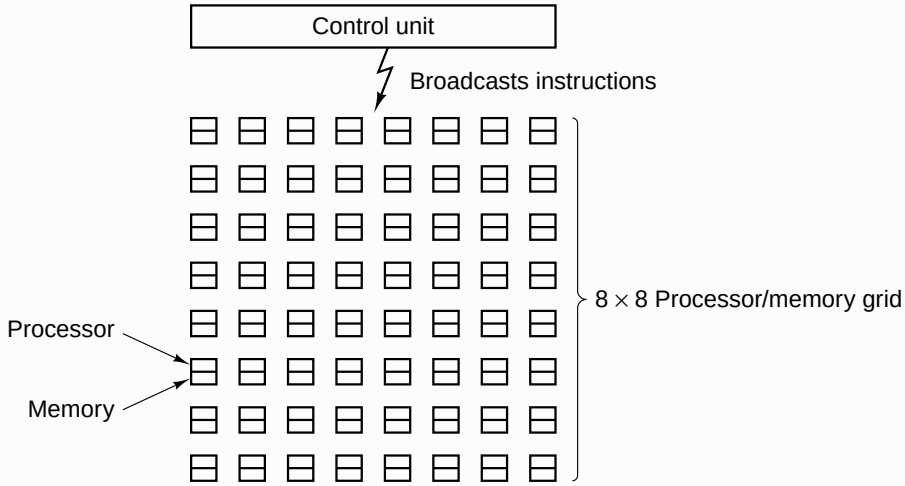
Classificazione di Flynn (1972)

- **SISD** Single Instruction Single Data (macchina di von Neumann)
- **SIMD** Single Instruction Multiple Date (Computer Vettoriali)
- **MIMD** Multiple Instruction Multiple Data (Multiprocessori e multicomputer)
- **MISD** Multiple Instruction Multiple Data (nessun esempio)

Una classificazione più fine



SIMD: Array processor



Più processori controllati da una unità: ILLIAC IV

GPU (General Purpose Unit)

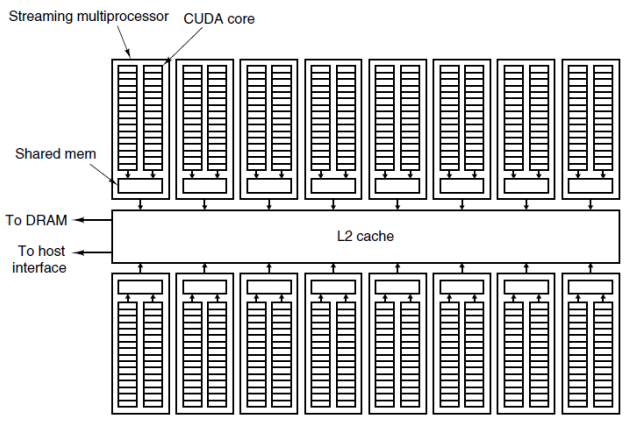
Le schede grafiche sono un implementazione moderna degli array processor.

- La computazione nelle GPU (Graphical process uniti) avviene secondo lo schema array processor:
- la stessa operazione eseguita da semplici core su dati diversi.
- struttura della memoria complessa:
  - ogni core ha una memoria privata,
  - gruppi di core condividono una memoria locale,
  - esiste una memoria comune a tutti i core.

GPGPU (General Purpose GPU)

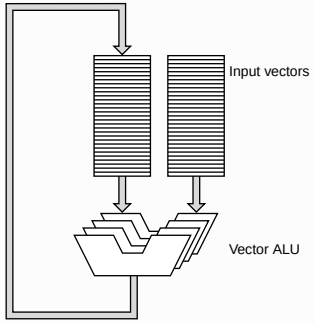
- Con i GPGPU si utilizzano le GPU per calcoli generici (non grafici),
  solo alcuni tipi di calcolo adatti a questo tipo di computazione,
  si sfruttano l'elevato numero di core (Nvidia Fermi - 512 Core).
- nuovi linguaggi di programmazione su GPGPU
  - CUDA (Nvidia)
  - OpenCL (Generico, usabile su più schede grafiche).

Nvidia Fermi



Gerarchia di memoria: locale al singolo core, condivisa tra gruppi di core, memoria comune.

SIMD: Vector processor



ALU e Registri operano su vettori: CRAY.

SIMD instruction

I processori attuali implementano istruzioni vettoriali.

Estensioni ai linguaggi macchina.

ISA x86:

- MMX (MultiMedia eXtension)
- SSE (Streaming SIMD Extension), ... , SSE4
- AVX (Advance Vector eXtension)

Utilizzati registri molto lunghi, centinaia di bit, contenenti vettori di dati (byte, half-word, word) su cui operare con istruzioni vettoriali.

SIMD instruction

Si distingue tra:

- istruzioni multimediali: lunghezza fissa.
- istruzioni vettoriali: insiemi di dati di lunghezza variabile, parametriche

Vantaggi:

- si sfrutta la legge di Moore (i tanti transistor a disposizione),
- istruzioni vettoriali permettono un codice più compatto.

Calcolatori MIMD

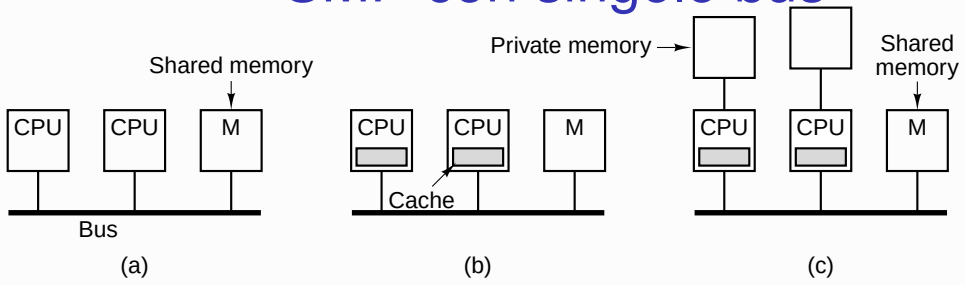
- multiprocessor
  - UMA Uniform Memory Access
  - NUMA Non Uniform Memory Access
  - COMA Cache Only Memory Access
- multicomputer
  - MPP Massive Parallel Processor
  - COW Cluster Of Workstation (NOW Network Of W.)

Multiprocessori UMA

Tutti i processori hanno stessi tempi di accesso alla memoria,

SMP (Symmetric Multi Processor): quando i processori hanno la stessa visione della memorie e dei dispositivi I/0:

SMP con singolo bus



- L'implementazione più semplice di un sistema multiprocessore.
- Bus limita il numero di CPU utilizzabili (16).
- Cache di grosse dimensioni per limitare l'accesso al bus.
- Necessità di mantenere la coerenza della cache.

Snooping cache

Le cache spia il traffico sul bus per garantire la coerenza.
Diverse implementazioni: Write through

- Read Miss – Lettura dalla memoria.
- Read Hit – Dati locali.
- Write Miss – Aggiorna memoria - Invalida altre linee.
- Write Hit – Aggiorna memoria e cache – Invalida altre linee.

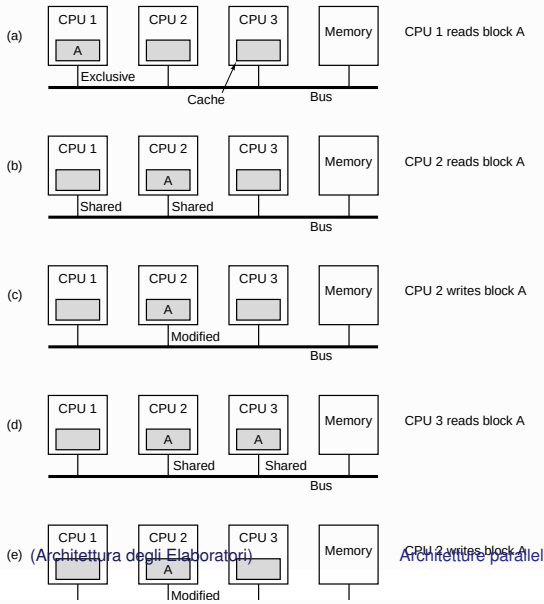
Molti accessi al bus: semplice ma poco efficiente.

Snooping cache: protocollo MESI

Protocollo write back.
Utilizzato nei processori Core.
Ogni linea di cache viene marcato in uno tra 4 modi:

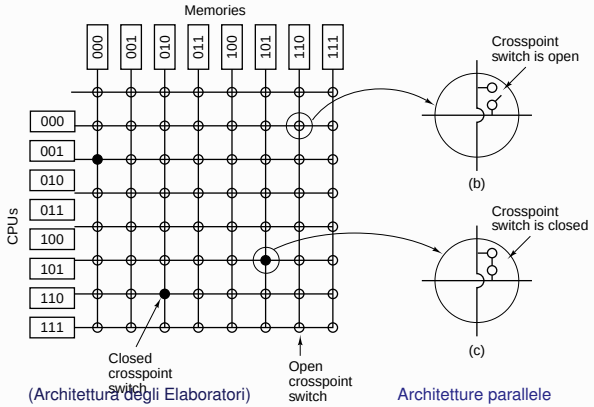
- Exclusive (sola copia in cache);
- Shared (linea presente in altre cache);
- Modified (linee cache diversa da quella in memoria);
- Invalid (linea non valida).

MESI, esempio



UMA con crossbar switch

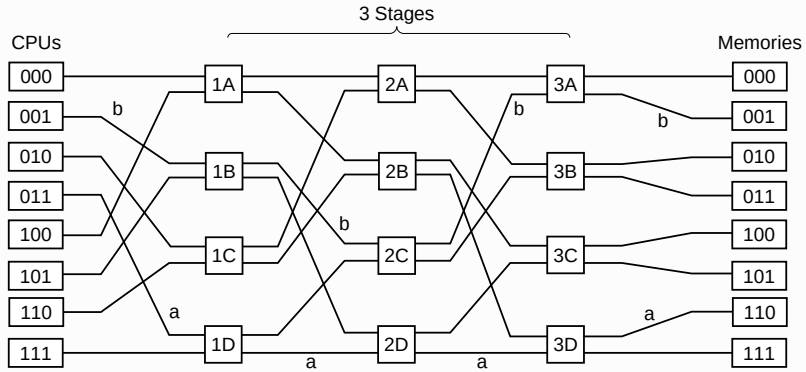
Processori e memorie connesse con un crossbar switch, (Sun Fire E25K).
Rete non bloccante.
Elevato numero di switch. ($\mathcal{O}(n^2)$)



UMA con reti a commutazione

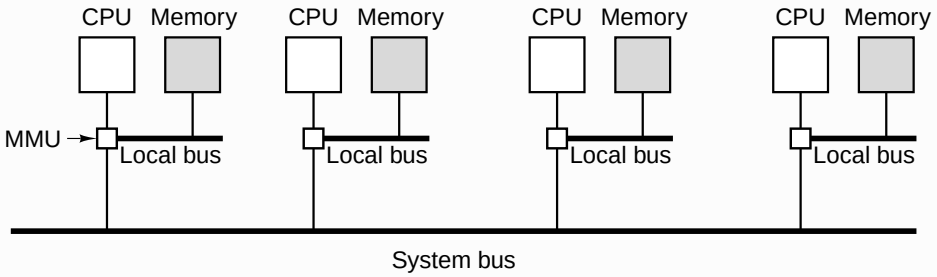
Rete Omega

- Rete bloccante.
- Limitato numero di switch. ($\mathcal{O}(n \log n)$)



Multiprocessori NUMA

Not Uniform Memory Access



Multiprocessori NUMA

- Memoria locale e remota
- Un unico spazio di indirizzamento
- maggiore scalabilità: superano il limite delle 100 CPU
- necessità di ridistribuire i dati

Hardware DSM Distributed Shared Memory.

Si dividono in:

- CC-NUMA presenza di caching – limita gli accessi remoti – directory (invece di snoooping cache) per garantire la coerenza
- NC-NUMA assenza di caching – aumentano gli accessi – paginazione sofisticata **page scanner**, per ridistribuire i dati
- COMA Cache Only Memory Access – la memoria locale è vista come una grande memoria cache – problemi recupero dati.

Multicomputer

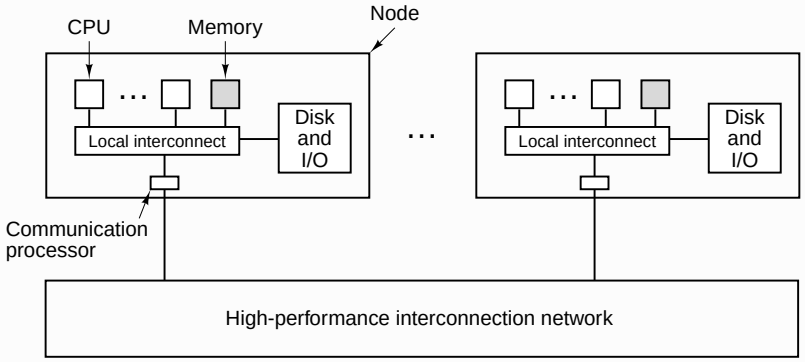
CPU con spazio privato di memoria, non direttamente accessibile
Vantaggi:

- maggiore scalabilità,
- efficienti nell'eseguire in parallelo processi indipendenti (o con poca dipendenza). Es. server web.

Caratteristiche:

- sistemi fault-tollerance,
- i singoli computer possono essere SMP con bus singolo.

Multicomputer



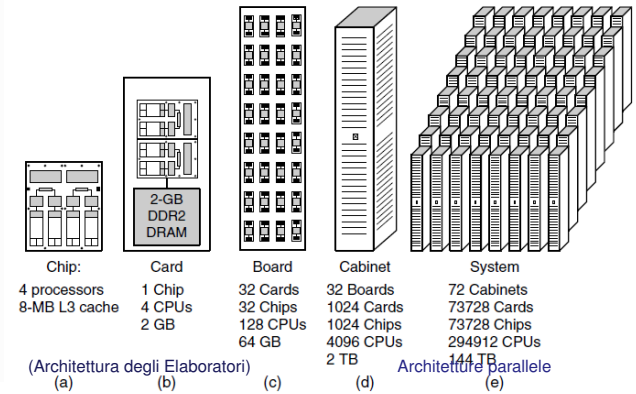
Si dividono in due categorie.

Massive Parallel Processor
MPP

- Gli attuali supercomputer
- Un elevato numero di processori standard
- Rete di interconnessione sofisticata
- Librerie software parallelo
- Fault tollerance: necessario per l'elevato numero di componenti
- Esempi: BlueGene: $\sim$: 400K CPU, Red Storm 10K CPU Opeteron.

BlueGene

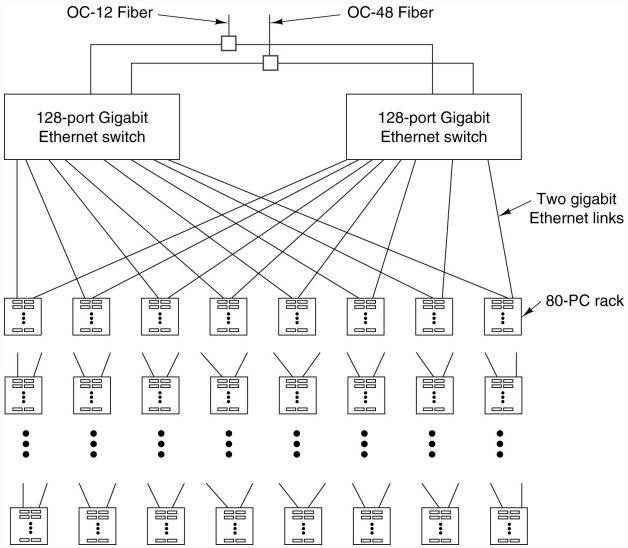
Progetto IBM.
Record di prestazioni teoriche ed effettive.
Campi d'applicazione: scacchi, simulazione: protein-unfolding, sistemi biologici (cervello, cuore), cosmologia.



Cluster of workstation
COW

- Rete di interconnessione standard: si sfruttano i progressi nelle reti.
- Dispositivi economici, facilmente assemblabili,
- possono raggiungere prestazioni paragonabile ai MPP.
- Centralizzati o decentralizzati.

Esempio: Cluster Google



Esempio: Cluster Google

Un numero limitato di datacenter (12) di grosse dimensioni.
Il progetto poco pubblicizzato.
Principi costruttivi:

- utilizzare componenti economici (desk-top, rete Ethernet) di largo uso, con il miglior rapporto: $\mathit{prestazioni}/(\mathit{prezzo} + \mathit{consumi})$; (consumi $\sim$100MW)
- l'affidabilità ottenuta attraverso ridondanza, la rottura di un componente non compromette il sistema.

Grid computing

Livello successivo di computazione parallela:
Computazione distribuita su calcolatori connessi via web.

- meccanismi di protezione,
- gestione della risorse di calcolo,
- distribuzione del carico,
- sistemi di calcolo non omogenei, definizione di uno standard comune.