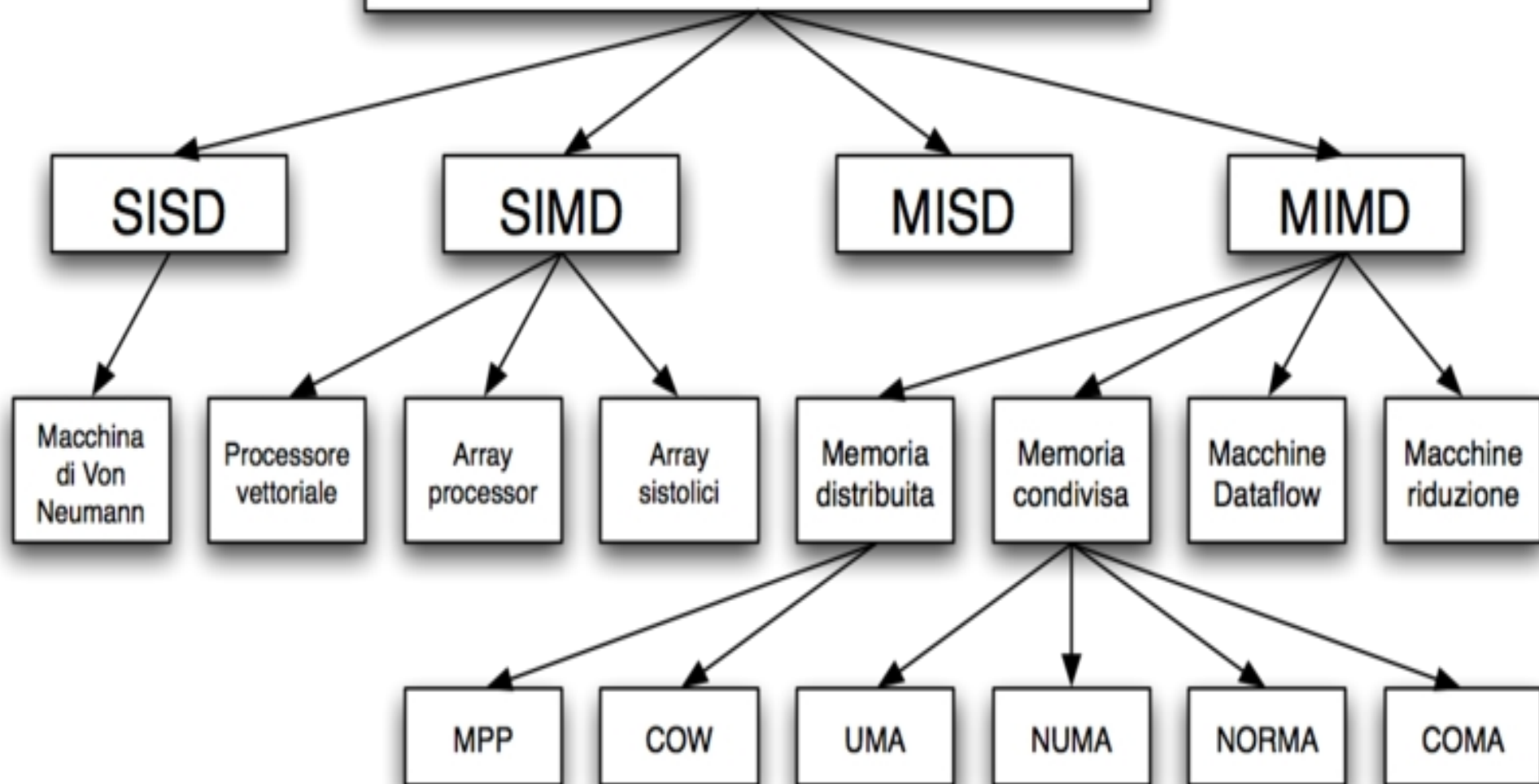


Architetture di calcolo



Architetture della memoria

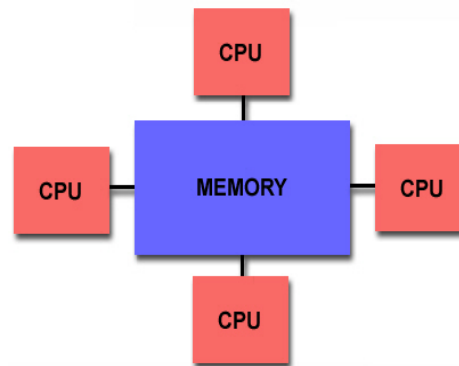
Un elemento determinante per disegnare una applicazione parallela e' l'architettura della memoria della macchina che abbiamo a disposizione. Rispetto all'architettura della memoria i calcolatori paralleli attuali possono essere

- . Shared Memory
- . Distributed Memory
- . Distributed Shared Memory

Shared Memory

Esistono diverse tipologie di computer paralleli Shared Memory ed hanno in comune la possibilita' per ogni processore di accedere a tutta la memoria come uno spazio globale di indirizzi

- . I processori operano indipendentemente, condividendo le stesse risorse di memoria
- . Un cambiamento in una locazione di memoria dovuto ad un processore e' visibile a tutti gli altri
- . Le macchine Shared Memory possono essere divise in due classi
 - . Uniform Memory Access (**UMA**)
 - . Non-Uniform Memory Access (**NUMA**)



Shared Memory

Uniform Memory Access (**UMA**):

- .Elaboratori costituiti da piu' processori identici e sovente chiamati Symmetric Multiprocessor (**SMP**)
- .I tempi d'accesso a tutta la memoria sono uguali per ogni processore
- .Tipicamente sono **CC-UMA** (Cache Coherent UMA)
 - .Cache Coherent significa che, se un processore aggiorna una locazione di memoria, il sistema provvede automaticamente ad invalidare eventuali locazioni di cache degli altri processori mappate sulla locazione di memoria appena modificata
 - .La Cache Coherency è mantenuta a livello hardware

Shared Memory

Non-Uniform Memory Access (**NUMA**):

- . Generalmente sono realizzate attraverso il collegamento di due o più SMP
- . Ogni SMP può accedere alla memoria degli altri SMP
- . I processori non hanno lo stesso tempo d'accesso a tutta la memoria (l'accesso alla memoria tramite link è più lento)
- . Se è mantenuta la cache coherency si parla di architetture CC-NUMA (Cache Coherent NUMA)

Shared Memory

Vantaggi:

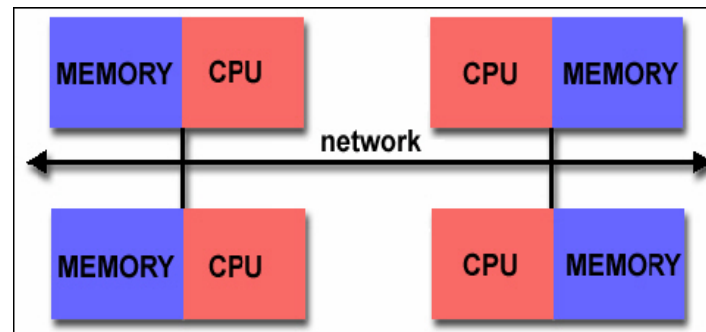
- .Lo spazio globale di indirizzi di memoria consente una programmazione piu' agevole rispetto agli accessi alla memoria
- .La condivisione di dati tra task e' resa agevole e veloce dalla prossimita' tra le CPU e la memoria

Svantaggi:

- .Scalabilita' limitata: aggiungere piu' CPU comporta una crescita geometrica del traffico tra le CPU e la memoria
- .la crescita del traffico tra la memoria e le CPU corrisponde ad una gestione molto complessa della gerarchia della memoria, per garantire la cache coherency
- .E' esclusiva responsabilita' del programmatore la sincronizzazione degli accessi alla memoria, per garantirne la correttezza
- .Costi: al crescere del numero di processori e' sempre piu' complesso e costoso disegnare e produrre sistemi shared memory

Distributed Memory

- Ogni processore dispone di un'area di memoria locale
- . I processori possono scambiare dati solo attraverso il network di comunicazione
 - . Non e' previsto un concetto di spazio di indirizzi globale tra i processori
 - . Poiche' ogni CPU dispone di una memoria locale, opera indipendentemente dalle altre. I cambiamenti in memoria locale non hanno effetti sulla memoria delle altre CPU, ovvero il concetto di cache coherency non e' applicabile



Distributed Memory

- . Nei casi in cui una CPU ha bisogno di dati residenti nella memoria di un'altro processore, e' responsabilita' del programmatore definire in modo esplicito come e quando i dati debbano essere comunicati
- . Al programmatore e' lasciata la responsabilita' di sincronizzare i task residenti su nodi diversi
- . Esistono varie tipologie di infrastrutture di comunicazione per consentire il trasferimento dei dati tra i nodi: da tecnologie proprietarie (Cray, IBM) a sistemi commodity (InfiniBand, Myrinet, QNet), alla ordinaria tecnologia Ethernet (Gigabit almeno! Comunque sia il problema maggiore in questi casi e' la latenza e non la banda)

Distributed Memory

Vantaggi:

- . La quantita' di memoria e' scalabile con il numero di processori, ovvero al crescere del numero dei processori la memoria disponibile cresce proporzionalmente
- . Ogni processore accede "rapidamente" alla propria memoria, senza interferenze e senza l'overhead dovuto al mantenimento della cache coherency.
- . Costi contenuti: puo' essere realizzata tramite processori ed infrastruttura di comunicazione disponibili sul mercato di massa (commodity off-the-shelf)

Svantaggi:

- . Il programmatore e' responsabile di tutti i dettagli legati alla comunicazione di dati tra i processori
- . Potrebbe essere complesso "mappare" il layout dei dati di codici esistenti (pensati per un unico spazio di indirizzamento) su un modello di memoria distribuita
- . Esiste una gerarchia di tempi d'accesso alla memoria, complessa quanto e piu' delle architetture NUMA

Distributed Shared Memory

Soluzione ibrida rispetto alle precedenti

- . Il singolo nodo e' un sistema Shared Memory (UMA o NUMA)
- . I nodi sono interconnessi tramite un network di comunicazione
- . La comunicazione tra i processori avviene
 - . Tramite un'area di memoria condivisa se sono sullo stesso nodo
 - . Tramite il network se sono su nodi diversi
- . I sistemi DSM in commercio sono di due tipologie, diverse per come e' presentata al programmatore la natura distribuita della memoria.
 - . NUMA: e' definito un unico spazio di indirizzi, con tempi d'accesso diversi in base alla "vicinanza" della particolare area di memoria al processore
 - . Cluster di SMP: ogni nodo ha il suo spazio di indirizzi e l'accesso alla memoria degli altri nodi avviene con operazioni di comunicazione esplicita