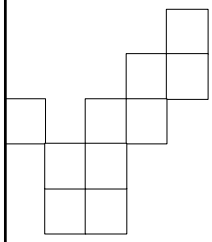


I sistemi multiprocessori

Corso di "Sistemi per Elaborazione dell'Informazione"
Prof. Bruno Carpentieri
A.A. 2004/2005

Loredana D'Arienzo
Maurizio Cembalo

Indice

- Sistemi Multiprocessore
  - Caratteristiche
  - Modelli di organizzazione
  - Coerenza della cache
- Cluster
  - Cluster VS Sistemi Multiprocessore

Le caratteristiche dei sistemi multiprocessore (1)

Un sistema multiprocessore è una macchina nella quale le applicazioni, per ottenere prestazioni sempre più elevate, hanno a disposizione più di un processore

Diverse sono le caratteristiche dei sistemi multiprocessore:

- Essi sono scalabili: Hw e sw sono progettati in modo da poter essere adattati ad un numero variabile di processori. Dato che il sw è scalabile alcuni sistemi multiprocessore possono funzionare anche in presenza di un malfuzionamento hw
- I sistemi multiprocessore possono raggiungere le migliori prestazioni in assoluto ed essere più veloci del più veloce sistema a processore singolo

Le caratteristiche dei sistemi multiprocessore (2)

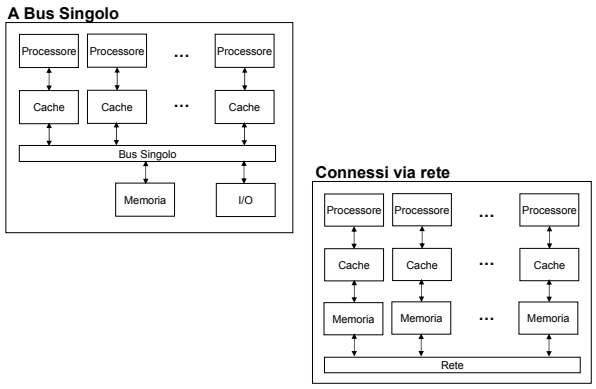
- Al momento sono i sistemi con il migliore rapporto costo/prestazioni
- In generale non è possibile gestire un carico di lavoro a suddivisione di tempo su un processore a singolo chip: costruire un sistema multiprocessore composto da più processori a singolo chip è più efficiente che costruire un processore singolo ad alte prestazioni

Alcune considerazioni

- Bisogna ottenere un buon livello di prestazioni ed efficienza altrimenti tanto varrebbe usare un solo processore dato che la programmazione è più semplice
- E' difficile scrivere programmi veloci per sistemi multiprocessore al crescere dei processori
- E' difficile scrivere programmi che sfruttano a pieno l'elaborazione parallela poiché i programmatori dovrebbero conoscere l'hardware davvero molto bene

I modelli di organizzazione

- I sistemi multiprocessore possono seguire due modelli base di organizzazione:
  - A bus singolo
  - Connessi via rete



Sistemi multiprocessore a bus singolo

- il mezzo di collegamento è unico ed è collocato tra i processori e la memoria. Il bus è usato per tutti gli accessi in memoria
- La comunicazione è effettuata mediante un ampio spazio di indirizzamento **condiviso**
- Occorre gestire i "conflitti" per l'accesso alla memoria e sincronizzare gli accessi alla stessa locazione da parte di più processori

Sistemi multiprocessore connessi via rete

- La memoria è ***fisicamente distribuita***, per supportare numeri più elevati di processori
- la memoria è connessa a ciascun processore ed il mezzo di collegamento (la rete) è fra i diversi nodi. Il sistema di interconnessione è coinvolto solo nella comunicazione fra processori diversi
- Ogni nodo accede direttamente alla propria parte di memoria; l'informazione condivisa deve essere *replicata* nelle memorie dei diversi nodi (la coerenza dei dati è garantita dal software – sistema operativo, sistema di DBM etc.)
- Punto critico: la banda della rete d'interconnessione

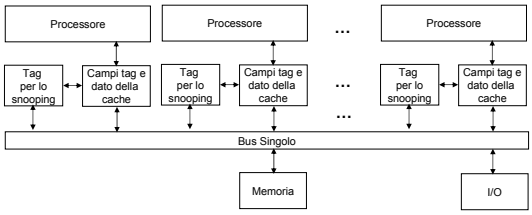
Coerenza della cache nei sistemi multiprocessore a bus singolo

- Problema della coerenza della cache: i processori multipli necessitano comunemente di copie degli stessi dati mantenute in cache diverse
- I processori devono avere accesso alla copia più recente, presente nella propria cache, quando leggono un dato, per cui tutti i processori devono ottenere il nuovo dato dopo una scrittura
- I protocolli che mantengono la coerenza nei sistemi multiprocessore vengono detti *Protocolli di coerenza della cache*: il più diffuso è snooping

Protocollo snooping

- Copie multiple dello stesso dato in più cache non sono un problema in caso di lettura; ma un processore deve avere accesso esclusivo in scrittura
- I protocolli di snooping devono quindi localizzare tutte le cache che condividono un oggetto che deve essere scritto; la conseguenza di un operazione di scrittura sui dati condivisi è l'invalidazione di tutte le altre copie oppure l'aggiornamento delle copie condivise con il nuovo valore che è stato scritto

Sistema multiprocessore a bus singolo che utilizza il protocollo di Snooping



Funzionamento dello snooping

- Al verificarsi di una situazione di miss in lettura tutte le cache controllano se possiedono una copia del blocco richiesto e agiscono in modo appropriato ad esempio fornendo il dato alla cache che ne era priva
- Nel caso delle operazioni di scrittura le cache controllano se hanno una copia del blocco ed agiscono in modo opportuno a seconda del tipo di protocollo di snooping utilizzato

Tipi di protocolli

- I protocolli di snooping sono di due tipi a seconda di ciò che avviene nel caso delle operazioni di scrittura:
  - Invalidazione in scrittura
  - Aggiornamento in scrittura

Invalidazione in scrittura

1. Il processore che esegue l'operazione di scrittura fa si che tutte le copie contenute nelle altre cache siano invalidate prima di cambiare la propria copia locale inviando sul bus un segnale di invalidazione
2. Tutte le cache controllano se possiedono una copia della parola scritta; se la posseggono invalidano il blocco
3. Il processore è libero di aggiornare i dati locali fino a quando un altro processore ne fa richiesta

Questo schema permette quindi operazioni di letture multiple ed operazioni di scritture singole

Aggiornamento in scrittura

1. il processore che esegue l'operazione di scrittura trasmette sul bus il nuovo dato
2. tutte le copie sono aggiornate con il nuovo valore

Invalidazione VS Aggiornamento

- Lo schema di aggiornamento trasmette in continuazione le operazioni di scrittura sui blocchi condivisi, mentre lo schema di invalidazione, cancella tutte le altre copie in modo che ci sia un'unica copia locale nelle operazioni di scritture successive
- Altra differenza è che il meccanismo di aggiornamento in scrittura usa il bus per aggiornare le copie dei dati condivisi in tutte le operazioni di scrittura, mentre l'invalidazione, usa il bus solo nel caso della prima operazione di scrittura per invalidare tutte le altre copie, e le scritture successive non generano attività sul bus
- Di conseguenza l'invalidazione in scrittura riduce la richiesta di banda del bus mentre l'aggiornamento in scrittura ha il vantaggio di far caricare prima i nuovi valori nella cache, il che può ridurre la latenza

Accesso concorrente al bus

- Che cosa succede se due processori cercano di scrivere nella stessa parola condivisa durante lo stesso ciclo di clock?

L'arbitro del bus decide quale processore ha per primo il controllo del bus, e questo processore invaliderà o aggiornerà la copia dell'altro processore, a seconda del protocollo usato. A questo punto il secondo processore potrà eseguire la sua scrittura

Esempio di un protocollo di invalidazione in scrittura (1)

- Ciascuno blocco della cache si trova in uno dei tre stati:
  - Solo lettura: il blocco di cache è pulito (non scritto) e può essere condiviso
  - Lettura/scrittura: il blocco di cache è sporco (scritto) e non può essere condiviso
  - Invalido: il blocco di cache non contiene dati validi

Esempio di un protocollo di invalidazione in scrittura (2)

- Le transizioni di stato nei blocchi di una cache si verificano in condizioni di miss in lettura e scrittura, e di hit in scrittura; gli hit in lettura non cambiano lo stato della cache
- Consideriamo le modalità di funzionamento del protocollo inerenti alle condizioni di miss in lettura e miss in scrittura

Esempio di un protocollo di invalidazione in scrittura (3)

- MISS IN LETTURA

1. Quando il processore si trova in questa condizione per quanto riguarda un blocco di cache, cambia lo stato di tale blocco in "sola lettura"
2. Acquisisce il controllo del bus e riscrive il vecchio blocco se questo si trovava nello stato di "Lettura/Scrittura" (sporco)
3. Tutte le cache degli altri processori osservano la situazione di miss in lettura per vedere se contengono il blocco in questione; se una di esse ne ha una copia che si trova nello stato di "Lettura/Scrittura", cambia lo stato del blocco in "Invalido"

La situazione di miss in lettura è quindi risolta eseguendo una lettura dalla memoria

Esempio di un protocollo di invalidazione in scrittura (4)

- MISS IN SCRITTURA

- Per scrivere un blocco, il processore acquisisce il controllo del bus, invia un segnale di invalidazione, scrive nel blocco e lo pone nello stato di "Lettura/Scrittura"
- Osservando il bus, tutte le altre cache controllano se hanno una copia del blocco, e se si la invalidano

Esistono molte varianti di questo protocollo per la coerenza della cache tra i quali il MESI, adottato dal Pentium Pro e dal PowerPC, protocollo di invalidazione in scrittura il cui nome è l'acronimo dei quattro possibili stati: Modified, Exclusive, Shared, Invalid

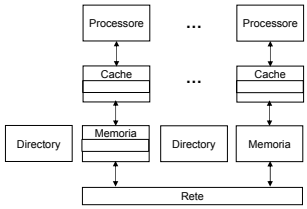
Coerenza della cache nei sistemi multiprocessore connessi via rete

- Un alternativa allo snooping del bus per garantire la coerenza della cache consiste nelle directory (elenchi)
- Nei protocolli basati sulle directory dal punto di vista logico esiste una sola directory che mantiene lo stato di ciascun blocco della memoria principale; le informazioni nella directory possono tenere traccia di quali cache hanno copie del blocco, se è sporco, e così via

Coerenza della cache nei sistemi multiprocessore connessi via rete (2)

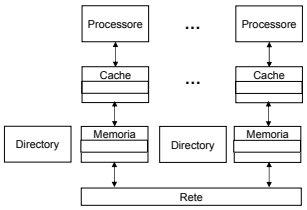
- Gli elementi delle directory possono essere distribuiti in modo che richieste diverse vadano a memorie diverse, riducendo così i conflitti e consentendo la scalabilità del progetto
- Le directory mantengono la caratteristica che lo stato di condivisione di un blocco è sempre in una sola locazione nota, rendendo plausibili i processori paralleli su larga scala

- La figura mostra la coerenza a livello di cache usando le directory in un sistema multiprocessore connesso via rete:



- Il dato originale è in memoria e le copie sono replicate solo nelle cache

- La figura mostra la coerenza a livello di memoria usando le directory in un sistema multiprocessore connesso via rete



- Le copie sono replicate nella memoria remota (rappresentata in colore azzurro) e nelle cache. Fin quando la memoria è coerente, i dati possono essere inseriti nelle cache senza problemi; se il dato in memoria è invalidato, allora i blocchi corrispondenti contenuti nelle cache devono essere ugualmente invalidati

Differenza rilevante con lo snooping

- meccanismo per scoprire quando si verifica una operazione di scrittura su un dato condiviso: invece di osservare il bus per controllare se ci sono richieste per cui è necessario aggiornare o invalidare la cache locale, il controllore della directory invia comandi espliciti a tutti i processori che mantengono una copia dei dati. Questi messaggi possono poi essere inviati sulla rete

Cluster

- Un cluster è un insieme di calcolatori indipendenti connessi da una rete locale, che sfruttano le reti locali per ottenere un elevata ampiezza della banda di comunicazione tra i calcolatori del cluster stesso
- Forniscono sistemi scalabili ed ampiamente disponibili, cioè sistemi il cui obbiettivo è sia contenere un numero di processori, di memorie e di dischi largamente variabile, sia essere disponibili 24 ore al giorno per 365 giorni l'anno

Cluster VS Sistemi Multiprocessore

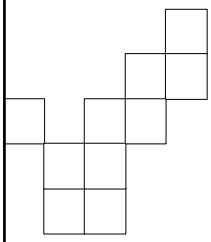
- I cluster sono collegati usando il bus di I/O del calcolatore, mentre i sistemi multiprocessore sono connessi tramite il bus della memoria. Quest'ultimo ha ampiezza di banda più elevata, permettendo ai sistemi multiprocessore di usare le connessioni di rete a velocità maggiore, e di avere meno conflitti con il traffico di I/O nel caso di applicazioni che ne fanno uso intensivo
- Suddivisione della memoria: un cluster di n macchine ha n memorie indipendenti ed n copie del sistema operativo, mentre un sistema multiprocessore con la memoria condivisa consente ad un singolo programma di usare quasi tutta la memoria del sistema
- Un programma sequenziale in un cluster ha quindi a disposizione una quantità di memoria pari a 1/n rispetto a quella a disposizione di un programma sequenziale su un sistema multiprocessore
- Il costo per l'amministrazione di un cluster di n macchine è più o meno lo stesso che per n macchine indipendenti, mentre il costo per l'amministrazione di un sistema multiprocessore con spazio di indirizzamento condiviso contenente n processori è più meno quello di una sola macchina

Cluster VS Sistemi Multiprocessore (2)

- In un cluster è più facile sostituire una macchina senza interrompere il funzionamento del sistema di quanto non avvenga per i sistemi multiprocessore
- Dato che il software del cluster costituisce uno strato al di sopra del sistema operativo di ciascun calcolatore, è molto più semplice disconnettere e sostituire una macchina non funzionante
- Dal momento che i cluster sono costruiti a partire da calcolatori indipendenti e da reti indipendenti e scalabili, questo isolamento rende anche più facile espandere il sistema senza interrompere l'applicazione che lavora sul cluster

Cluster VS Sistemi Multiprocessore (3)

- Per combattere la debolezza dei sistemi multiprocessore, nei cluster i progettisti hardware e gli sviluppatori dei sistemi operativi cercano di offrire la possibilità di usare più sistemi operativi su porzioni diverse della stessa macchina, in modo tale che un nodo possa guastarsi o essere aggiornato senza interrompere il funzionamento dell'intero sistema
- Poiché l'amministrazione del sistema e i limiti della dimensione della memoria sono legati in modo approssimativamente lineare al numero delle macchine indipendenti, c'è la tendenza a ridurre i problemi dei cluster costruendo cluster di SMP di piccole dimensioni
  Ad esempio un cluster di 32 processori potrebbe essere costruito a partire da 8 SMP a quattro vie oppure da 4 SMP a otto vie



I sistemi multiprocessori

parte seconda

Corso di "Sistemi per Elaborazione dell'Informazione"
Prof. Bruno Carpentieri
A.A. 2004/2005

Loredana D'Arienzo
Maurizio Cembalo

Indice

- Caratteristiche delle reti d'interconnessione
- Topologie delle reti d'interconnessione
  - Metriche
- Classificazione di Flynn
  - Macchina SISD / SIMD / MISD / MIMD
  - Dettagli sulle macchine SISD
  - Calcolatori vettoriali

Componenti di un calcolatore parallelo

- Unità di calcolo (processori)
- Unità di memoria
- Rete di interconnessione

Rete di interconnessione

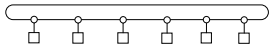
- Equivalente sofisticato dei bus
  Sono la componente fondamentale: a volte rappresenta la componente più costosa del calcolatore
- Una rete è composta da:
  - Link (connessioni - cavi - bus) paralleli o seriali
  - Switch (instradatori - commutatori)
  - Interfacce (Processori - rete - memorie)

Caratteristiche delle reti

- Il modo più diretto di connettere i nodi processore-memoria è predisporre un collegamento dedicato fra ciascun nodo
- Il **costo** della rete include il numero di switch, il numero di collegamenti su uno switch per le connessioni di rete, l'ampiezza del collegamento e la lunghezza dei collegamenti
- Per perseguire buone **prestazioni** occorre massimizzare l'ampiezza di banda ed evitare colli di bottiglia
- Le reti devono essere **tolleranti ai guasti**

Topologie delle reti

- Le reti sono disegnate come dei grafi, in cui ciascun arco del grafo rappresenta un collegamento; il nodo processore-memoria è rappresentato da un quadratino nero e lo switch da un cerchietto colorato
- Tutti i collegamenti sono bidirezionali; tutte le reti consistono di switch i cui collegamenti vanno dai nodi processore-memoria ad altri switch
- Il primo miglioramento rispetto al bus è una rete ad anello che connette una sequenza di nodi:



Dato che non tutti i nodi sono connessi in modo diretto, alcuni messaggi devono attraversare dei nodi intermedi fino ad arrivare alla destinazione finale.
A differenza del bus, l'anello può effettuare più trasferimenti in modo simultaneo.

Metriche per le topologie di rete

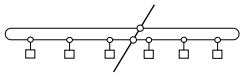
- Essendo possibile scegliere tra numerose topologie di rete è necessario definire delle metriche per misurare le prestazioni
- Di queste, due sono particolarmente diffuse:
  - la *banda totale della rete*
  - la *banda di bisezione*

Banda totale della rete

- Rappresenta la banda di ciascun collegamento moltiplicata per il numero di collegamenti
- Questa metrica rappresenta il caso migliore: per la rete ad anello con P processori, la banda totale della rete sarebbe P volte la banda di un collegamento
- la banda totale della rete nel caso di un bus è semplicemente la banda del bus stesso o in altre parole una volta la banda del collegamento

Banda di bisezione

- Rappresenta una metrica vicina al caso peggiore
- Viene calcolata suddividendo la macchina in due parti, ciascuna comprendente la metà dei nodi, e sommando la banda dei collegamenti che attraversano la linea immaginaria di suddivisione



- La banda di bisezione di una rete ad anello è due volte la banda del collegamento, e quella del bus è una volta tale banda
- se un singolo collegamento è veloce quanto un bus, l'anello è due volte più veloce del bus nel caso peggiore, ma è P volte più veloce nel caso migliore

Banda di bisezione (2)

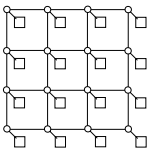
- Dato che alcune topologie di rete non sono simmetriche possono esserci dei problemi per stabilire dove far passare la linea di suddivisione immaginaria
- Bisogna scegliere la suddivisione che conduce alle prestazioni di rete peggiori essendo la metrica vicino al caso peggiore
- Consiste nel calcolare tutte le possibili bande di bisezione e nello scegliere la più bassa, infatti i programmi per l'elaborazione parallela sono spesso limitati dall'anello più debole della catena della comunicazione

Rete completamente connessa

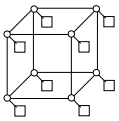
- All'estremo opposto dell'anello si trova la *rete completamente connessa*, in cui ciascun processore ha un collegamento bidirezionale con tutti gli altri processori
  - Banda totale della rete: Px(P-1)/2 volte quella di un singolo collegamento
  - Banda di bisezione: $(P/2)^2$ volte la banda di un collegamento.
- Il miglioramento di prestazioni di una rete completamente connessa è controbilanciato da un aumento enorme dei costi

inventare nuove topologie di rete che hanno caratteristiche intermedie fra il costo delle reti ad anello e le prestazioni delle reti completamente connesse

Topologie di rete più utilizzate



Griglia 2-D o maglia di 16 nodi



n-cubo di 8 nodi

- le macchine reali aggiungono frequentemente dei collegamenti a queste semplici topologie per migliorarne le prestazioni e l'affidabilità

Reti multistadio

- Anziché sistemare un processore in ciascun nodo della rete si può lasciare soltanto uno switch in alcuni di questi nodi; gli switch sono più piccoli dei nodi processore-memoria-switch, per cui possono essere impaccati in modo più denso, diminuendo le distanze e migliorando le prestazioni
- Queste reti sono spesso chiamate *reti multistadio*, per riflettere il fatto che un messaggio deve attraversare più stadi. Le tipologie di rete multistadio sono numerose quante quelle a stadio singolo

Organizzazione delle reti multistadio

- Le due organizzazioni multistadio più diffuse sono:
  - la rete completamente connessa detta *rete crossbar* che consente a ciascun nodo di comunicare con qualunque altro nodo
  - la *rete omega* che usa meno hardware di una rete crossbar ma in cui si possono verificare situazioni di conflitto fra messaggi, a seconda della struttura della comunicazione

Implementare le topologie di rete: considerazioni pratiche

- Maggiore è la distanza di ciascun collegamento più è costoso utilizzare il collegamento a frequenze di clock elevate
- Distanze più brevi rendono più semplice assegnare più conduttori al collegamento, dato che in un chip la capacità richiesta per pilotare più conduttori è minore se i conduttori sono corti; i conduttori più corti infine sono meno costosi dei conduttori lunghi
- Le rappresentazioni tridimensionali devono essere implementate su chip e piastre essenzialmente bidimensionali. Topologie che disegnate su una lavagna appaiono eleganti possono diventare complicate e poco eleganti quando realizzate praticamente con chip, cavi, piastre e scatole

Considerazioni sull'assenza di una topologia standard

- La mancanza di una topologia standard non rappresenta uno ostacolo per la scrittura di programmi portabili sui sistemi paralleli
- I seguenti elementi se combinati riducono l'importanza degli algoritmi basati su una specifica topologia:
  - La mancanza di una topologia standard e l'importanza di programmi portabili per processori paralleli
  - l'elevato costo della comunicazione, che fa sì che il tempo di latenza sia virtualmente lo stesso per tutti i messaggi, indipendentemente dalla distanza tra i nodi connessi
  - La necessità di operare anche in presenza di guasti nelle connessioni e nei nodi

Classificazione di Flynn

- Poiché si può parlare di parallelismo a diversi livelli, è utile disporre di una classificazione delle varie possibilità esistenti
- Nel 1966 Flynn propose un semplice modello di classificazione per i calcolatori ancora oggi utilizzato
  Osservando le componenti elementari di un elaboratore, egli propose di contare il numero di flussi paralleli di istruzioni e di dati e di caratterizzare i calcolatori sulla base di questo valore:
  - Un flusso di istruzioni, un flusso di dati (SISD)
  - Un flusso di istruzioni, più flussi di dato (SIMD)
  - Più flussi di istruzioni, un flusso di dati (MISD)
  - Più flussi di istruzioni, più flussi di dati (MIMD)
- Alcuni sistemi di elaborazione rappresentano degli ibridi rispetto a queste categorie

Macchina SISD

- *Single instruction stream, single data stream* (SISD)
- Macchina di Von Neumann
  - monoprocessore, in grado di eseguire una sola istruzione per volta su un dato di tipo scalare
- Macchina con flusso singolo di istruzioni (programma) e flusso singolo di dati (sequenza di dati)
  - Elaborazione strettamente sequenziale

Macchina SIMD

- *Single instruction stream, multiple data streams* (SIMD)
- Macchina con un'unica unità di controllo e più elementi di calcolo identici.
  - una stessa istruzione viene eseguita da più processori usando diversi flussi di dati
- Ogni processore ha la propria memoria dati; c'è una sola memoria istruzioni e un solo processore di controllo. I processori multimediali mostrano una forma limitata di parallelismo SIMD; le architetture vettoriali sono la classe più ampia

Macchina MISD

- *Multiple instruction streams, single data stream* (MISD)
- non si è mai costruito nessun multiprocessore commerciale di questo tipo
- approssimato da qualche "stream processor" molto speciale (più unità funzionali operano su un unico flusso di dati)

Macchina MIMD

- *Multiple instruction streams, multiple data streams* (MIMD)
- Macchina con più processori che operano in modo asincrono: eseguono procedure diverse in parallelo (differenti istruzioni su dati differenti)
- ogni processore legge le proprie istruzioni e opera sui propri dati. I processori spesso sono microprocessori commerciali standard

Dettagli sui calcolatori SIMD

- I calcolatori SIMD lavorano su vettori di dati. Quando ad esempio una singola istruzione SIMD somma 64 numeri, la circuiteria SIMD invia 64 flussi di dati a 64 ALU in modo da calcolare 64 somme in un solo colpo di clock
- Tutte le unità di elaborazione parallela sono sincronizzate ed eseguono una stessa istruzione che viene indirizzata da uno stesso Program Counter
- Dal punto di vista del programmatore, questo modello è simile a quello dei sistemi SISD: anche se ciascuna unità esegue la stessa istruzione, ciascuna unità di esecuzione possiede i propri registri di indirizzo, così che ciascuna unità può utilizzare indirizzi di dato diversi

Vantaggi dei sistemi SIMD

- I sistemi SIMD permettono di ammortizzare il costo dell'unità di controllo su decine di unità di esecuzione.
- Ridotta dimensione della memoria di programma, dal momento che un sistema SIMD richiede una sola copia del codice in corso di esecuzione, mentre i sistemi MIMD possono richiedere una copia in ciascun processore. Il meccanismo della memoria virtuale e le maggiori capacità dei chip DRAM hanno ridotto l'importanza di questo vantaggio.
- I calcolatori SIMD in realtà possiedono un misto di istruzioni SISD e SIMD. Normalmente vi è un calcolatore host di tipo di SISD che esegue le operazioni sequenziali quali i salti o il calcolo degli indirizzi; le istruzioni SIMD sono invece inviate a tutte le unità di esecuzione, ciascuna dotata del proprio insieme di registri e della propria memoria.

Prestazioni dei sistemi SIMD

- I sistemi SIMD lavorano al meglio quando hanno a che fare con vettori all'interno di cicli *for:* affinchè il parallelismo sia efficace bisogna che vi siano grandi quantità di dati da elaborare
- I sistemi SIMD si trovano in difficoltà con i costrutti *case* o *switch*, nei quali ciascuna unità di esecuzione deve eseguire una diversa operazione sui propri dati locali, a seconda del valore di tali dati.
  - Le unità di esecuzione con i dati sbagliati sono disabilitate, mentre quelle con i dati corretti possono continuare l'elaborazione.
  - Si ha un calo di prestazioni pari a *n* volte, dove *n* è il numero di clausole del costrutto.

Prestazioni dei sistemi SIMD (2)

- Un parametro cruciale negli elaboratori SIMD è dato dalle prestazioni del processore rispetto al numero di processori.
  - Il sistema Connection Machine 2 (CM-2) ad esempio si basa su 65536 processori con ampiezza di banda pari ad un bit, mentre il sistema Illiac IV ha 64 processori con parallelismo pari a 64 bit.
  - Il sistema SIMD più conosciuto è l'Illiac IV.

Calcolatori vettoriali

- Un modello connesso al SIMD è quello dell'*elaborazione vettoriale*. Si tratta di un'architettura oggi ben assestato che è considerevolmente più utilizzato rispetto al modello SIMD
- I processori vettoriali dispongono di operazioni di alto livello in grado di lavorare su schiere lineari di numeri, o vettori.
  - Un esempio di operazione su vettori è

    A=BxC

    dove A, B e C sono vettori di 64 elementi corrispondenti ciascuno ad un numero in virgola mobile di 64 bit.
- I sistemi SIMD hanno delle istruzioni analoghe; la differenza risiede nel fatto che i processori vettoriali si basano su unità funzionali con pipeline che operano tipicamente su pochi elementi per ciascun colpo di clock, mentre un sistema SIMD lavora generalmente su tutti gli elementi in una volta sola.

Vantaggi dei calcolatori vettoriali

- Ciascun risultato è indipendente dai risultati precedenti, e questo consente di utilizzare pipeline più profonde e frequenze di clock più elevate
- Ciascuna istruzione vettoriale esegue una considerevole mole di lavoro, e questo comporta un minor numero di operazioni di prelievo delle istruzioni, un numero minore di istruzioni di salto, e quindi un numero minore di salti con previsione errata
- Le istruzioni vettoriali accedono alla memoria in blocchi, e questo permette di ammortizzare il tempo di accesso alla memoria su un numero di elementi elevato (ad esempio 64)
- Le istruzioni vettoriali accedono alla memoria secondo schemi noti, e questo permette a più banchi di memoria di fornire simultaneamente gli operandi

Considerazioni sui calcolatori vettoriali

- I processori vettoriali non necessitano di disporre di cache di dato con frequenze di hit elevate per raggiungere buone prestazioni
- Essi tendono a basarsi su memorie principali a bassa latenza, spesso costituite di SRAM, e ad avere un numero elevato di banchi di memoria (ad esempio 1024) per raggiungere un'elevata ampiezza di banda verso la memoria
- Per ottenere prestazioni ancora più elevate tutti i supercalcolatori vettoriali dispongono di processori multipli