

Corsi speciali abilitanti

Storia dell'informatica e del calcolo automatico

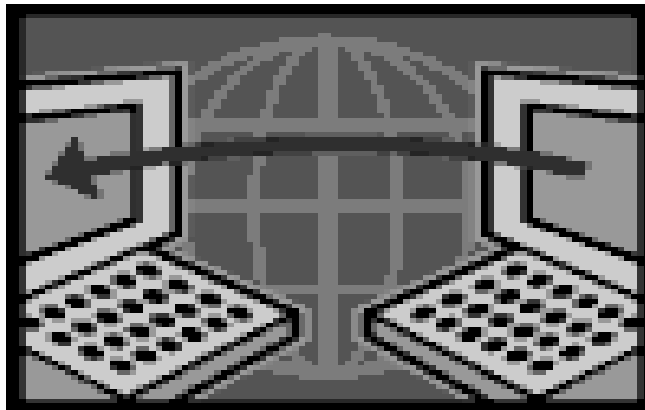
Docente corso

Murano Aniello

Docente abilitante

D'Avino Assunta

Sistemi di calcolo parallelo



- Negli ultimi anni sono state delineate le principali caratteristiche architettoniche delle macchine parallele conformemente ai modelli teorici di calcolo parallelo e alle metriche usate per misurarne le prestazioni

- Sebbene attualmente vi siano macchine parallele che vengono impiegate come macchine dedicate per supportare applicazioni specifiche (trattamento di immagini, robotica, visione, ecc.), è sempre più diffusa la necessità di avere a disposizione sistemi di tipo *general-purpose*. Per soddisfare questa richiesta è necessario un modello di macchina astratta standard che svolga il ruolo che il modello di Von Neumann ha svolto per gli elaboratori sequenziali.

- La più famosa e accettata classificazione delle architetture per i sistemi paralleli è quella proposta da M.J.Flynn.
- Secondo questa classificazione, le due più importanti caratteristiche di un elaboratore sono: il numero di flussi di istruzioni che esso può processare ad ogni istante, e il numero di flussi di dati su cui esso può operare simultaneamente.

- Combinando queste due caratteristiche è possibile ottenere le seguenti quattro classi architetturali:
- **SISD** (Single Instruction stream – Single Data stream)
- **SIMD** (Single Instruction stream – Multiple Data stream)
- **MISD** (Multiple Instruction stream – Single Data stream)
- **MIMD** (Multiple Instruction stream – Multiple Data stream)

- La classe SISD comprende l'architettura tradizionale di Von Neumann che è quella usata da tutti i calcolatori convenzionali, in cui il singolo processore obbedisce ad un singolo flusso di istruzioni (programma sequenziale) ed esegue queste istruzioni ogni volta su un singolo flusso di dati.

- Alla classe SIMD appartengono le architetture composte da molte unità di elaborazione che eseguono contemporaneamente la stessa istruzione ma lavorano su insiemi di dati diversi.
- Generalmente, il modo di implementare le architetture SIMD è quello di avere un processore principale che invia le istruzioni da eseguire contemporaneamente ad un insieme di elementi di elaborazione che provvedono ad eseguirle.

- Il modello rappresentato dalla classe MIMD, in cui più processi, eventualmente creati dinamicamente, sono in esecuzione contemporaneamente su più processori ed utilizzano dati propri o condivisi, rappresenta una evoluzione della classe SISD.
- Infatti, la realizzazione di queste architetture avviene attraverso l'interconnessione di un numero elevato di elaboratori di tipo convenzionale.
- I sistemi con architettura MIMD sono oggi fra quelli più studiati e si può presumere che essi rappresentino il punto di partenza per la costruzione di macchine parallele di tipo *general-purpose*.

- Sebbene la tassonomia di Flynn sia in grado di rappresentare alcuni aspetti fondamentali nella maggior parte delle architetture parallele, essa non è in grado di esplicitare pienamente tutte le caratteristiche interessanti per un programmatore.
- Infatti, essa non è in grado di distinguere fra architetture a memoria condivisa e architetture a memoria distribuita.
- Inoltre, in essa non trovano adeguata collocazione i calcolatori vettoriali, le macchine data-flow e quelle a riduzione che sono utilizzate come architetture parallele per la implementazione di linguaggi funzionali.

In particolare è possibile introdurre una ulteriore sottoclassificazione:

- SIMD
- Processori vettoriali
- Array processor
- Array sistolici
- MIMD
- Sistemi a memoria distribuita
- Sistemi a memoria condivisa
- Macchine Data-Flow
- Macchine a riduzione

- Le architetture SIMD utilizzano un modello di computazione in parallelo di tipo *sincrono*.
- Questo modello permette di coordinare l'esecuzione di più operazioni concorrenti attraverso intervalli di tempo che hanno una durata fissa, pari al tempo necessario per eseguire una operazione

- Il modello prevede che una computazione sia suddivisa in più fasi e che all'interno di ogni fase le computazioni possano essere partizionate per esplicitare parallelismo di tipo temporale o spaziale.

- Le architetture MIMD sono caratterizzate da una grande flessibilità che permette a questi sistemi di supportare su una stessa piattaforma hardware diversi modelli computazionali.
- Il modello architetturale MIMD può essere suddiviso in sistemi a memoria condivisa detti multiprocessor e sistemi a memoria distribuita conosciuti come multicomputer.

- A livello architetturale, i processori del sistema (nodi) cooperano secondo un modello asincrono. Secondo questo modello i vari nodi possono eseguire, in maniera autonoma, più flussi di istruzioni (processi) che usano dati locali o condivisi.
- I processi su ogni nodo vengono eseguiti facendo riferimento al tempo locale del processore.
- L'assenza di un tempo globale fa sì che, a differenza del modello sincrono, sia necessario disporre di meccanismi di comunicazione e sincronizzazione per consentire ai vari processi di scambiarsi informazioni sullo stato del sistema.

- Se si intende realizzare un modello computazionale asincrono la comunicazione fra processi dovrà avere una semantica non bloccante sia per le primitive di output sia per quelle in input.
- Un messaggio inviato da un processo è depositato in un buffer, se la primitiva corrispondente non è pronta a ricevere il dato.
- Il processo che ha inviato il messaggio continua l'elaborazione e successivamente gli verrà segnalato che il messaggio è stato ricevuto.
- Questi meccanismi di comunicazione riducono la sincronizzazione e favoriscono una esecuzione più parallela dei processi e una loro maggiore indipendenza.

- Nel caso di architetture MIMD a memoria condivisa è possibile emulare un modello sincrono o asincrono utilizzando un linguaggio concorrente che utilizzi un modello di cooperazione a memoria globale e disponga di costrutti di sincronizzazione del tipo semafori o monitor.

- Questi modelli per la cooperazione fra processi permettono di utilizzare macchine MIMD sia come paradigmi di programmazione a parallelismo esplicito che implicito.
- Nel caso esplicito, le attività concorrenti sono espresse direttamente come processi del linguaggio concorrente.
- Nel caso implicito, il programma sorgente è trasformato, mediante compilatori, in una rete di processi cooperanti.

- I multicomputer sono programmati attraverso il paradigma di scambio messaggi, attraverso una rete di interconnessione, mentre i multiprocessori usano il modello a memoria condivisa.
- Uno dei limiti principali delle architetture di tipo multiprocessor è quello di non poter essere costituite, a causa dei problemi di accesso in memoria e dei ritardi introdotti dalla rete, da molti processori, mostrando così una bassa scalabilità.

- I multicomputer sono sistemi caratterizzati da un numero elevato (dalle centinaia alle migliaia) di elaboratori (processore e memoria) ad altissima scala di integrazione, interconnessi da strutture regolari.
- Ogni elaboratore è dotato di un insieme di elementi di connessione (link) che gli permettono di collegarsi ad altri elaboratori secondo strutture statiche o dinamiche di tipo punto-a-punto.
- La struttura di interconnessione è scelta con l'obiettivo di mantenere piccola la distanza fra due nodi qualsiasi e di avere un basso numero di link per processore.

- Se gli algoritmi utilizzati impongono che la maggior parte degli accessi avvenga su dati locali e i processi hanno un comportamento indipendente, allora il carico sulla rete è notevolmente ridotto e le performance del sistema diventano elevate.
- Non sempre gli algoritmi sono caratterizzati da una elevata località. In questi casi, il sistema utilizza pesantemente la rete di comunicazione e necessita di algoritmi di instradamento (routing) dei messaggi per garantire una completa connettività logica fra i nodi.
- Gli algoritmi di routing utilizzati sono generalmente dinamici, cioè decidono il percorso a tempo di esecuzione e consentono di bilanciare il carico sui vari link in modo da evitare fenomeni di saturazione.

- Per ottenere elevate prestazioni in termini di efficienza e scalabilità è necessario che gli algoritmi da eseguire su un multicomputer siano progettati effettuando un adeguato bilanciamento fra il tempo di elaborazione e il tempo per la consegna dei messaggi (granularità dei processi) e definendo opportune strategie per l'allocazione delle attività ai vari nodi di elaborazione mantenendone bilanciato il carico di elaborazione.

- Le macchine data-flow e a riduzione sono sistemi caratterizzati da un nuovo approccio alla programmazione parallela.
- Il modello architetturale usato è sostanzialmente il modello MIMD (cooperazione asincrona fra i nodi ed esecuzione di attività concorrenti), ma il nuovo paradigma computazionale che esse usano è in grado di fornire una visione più astratta dell'architettura.

- Diversamente dalla macchina di Von Neumann, in cui le istruzioni sono eseguite sequenzialmente controllate da un program counter, queste architetture basano il loro funzionamento su due modelli computazionali: *data-driven* e *demand-driven*.
- Il modello data-driven prevede che una istruzione possa essere eseguita solo se tutti gli operandi che essa usa sono disponibili.
- Nel modello demand-driven è la richiesta del risultato che fa partire l'esecuzione dell'istruzione che lo deve calcolare.
- Entrambi i modelli non utilizzano un program counter e l'esecuzione di una istruzione avviene solo in base alla disponibilità dei dati.
- Le macchine data-flow utilizzano un modello data-driven e le macchine a riduzione un modello demand-driven.

- Nelle architetture data-flow i meccanismi di controllo della sequenza delle istruzioni tipici della programmazione imperativa non sono presenti. Esse vengono utilizzate per l'esecuzione di programmi funzionali o logici in cui il modello astratto è espresso attraverso un modello data-driven.
- Questo modello computazionale può essere assimilato ad un modello di computazione concorrente asincrona a scambio messaggi in cui i nodi possono avere granularità pari a quella dei processi o a quella di una singola istruzione.
- Ogni istruzione può essere implementata come un *template*, che è composto da un campo operatore, una memoria per ricevere gli operandi, e un campo con l'indicazione dei destinatari a cui spedire il risultato.

- Per far partire l'esecuzione tutti i valori degli operandi devono essere ricevuti nelle posizioni ad esse riservate nel template.
- I grafi data-flow sono in grado di esplicitare due forme di parallelismo.
- La prima forma permette a due nodi di essere eseguiti in parallelo se non vi è dipendenza fra i dati (parallelismo spaziale).
- La seconda forma è ottenuta dalle computazioni pipeline indipendenti che sono presenti nel grafo (parallelismo temporale).

- Le macchine a riduzione utilizzano un modello demand-driven per controllare il flusso della computazione.
- Il modello prevede che una istruzione venga abilitata per l'esecuzione se i risultati che essa produce sono necessari come operandi per un'altra istruzione che è già abilitata.

- Uno dei principali requisiti da soddisfare affinché l'elaborazione parallela diventi una tecnologia su cui basare le applicazioni del futuro, è quello di disporre di un modello standard di macchina astratta, simile al modello di Von Neumann per l'elaborazione sequenziale, in modo da separare gli aspetti implementativi software da quelli hardware.
- Quello che serve è un modello astratto su cui compilare efficientemente i linguaggi di alto livello e che possa essere implementato efficientemente in hardware, in modo che sia possibile eseguire un programma con la stessa efficienza su macchine parallele diverse.

- Un modello teorico consente di valutare i limiti teorici delle prestazioni delle macchine parallele e di effettuare un'analisi della scalabilità e dell'efficienza degli algoritmi paralleli.
- I modelli teorici più conosciuti sono:
- Il modello PRAM (Parallel Random Access Machine)
- Il modello Spatial Machines basato sugli automi cellulari
- Il modello BSP (Bulk Synchronous Parallel)
- Il modello LogP

bibliografia

- Università “La Sapienza” Roma
- D’Avino Assunta
- Internet

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.