

Introduzione ai Sistemi di elaborazione Concorrenti e Paralleli

Concetti fondamentali

Concorrenza&Parallelismo

- Parallelismo
- Concorrenza
- Concorrenza vs. parallelismo

Modelli di interazione di processi concorrenti/paralleli:

- Modello a memoria comune
- Modello a scambio di messaggi

Modelli architetturali

Concorrenza

- **Sistema concorrente:**

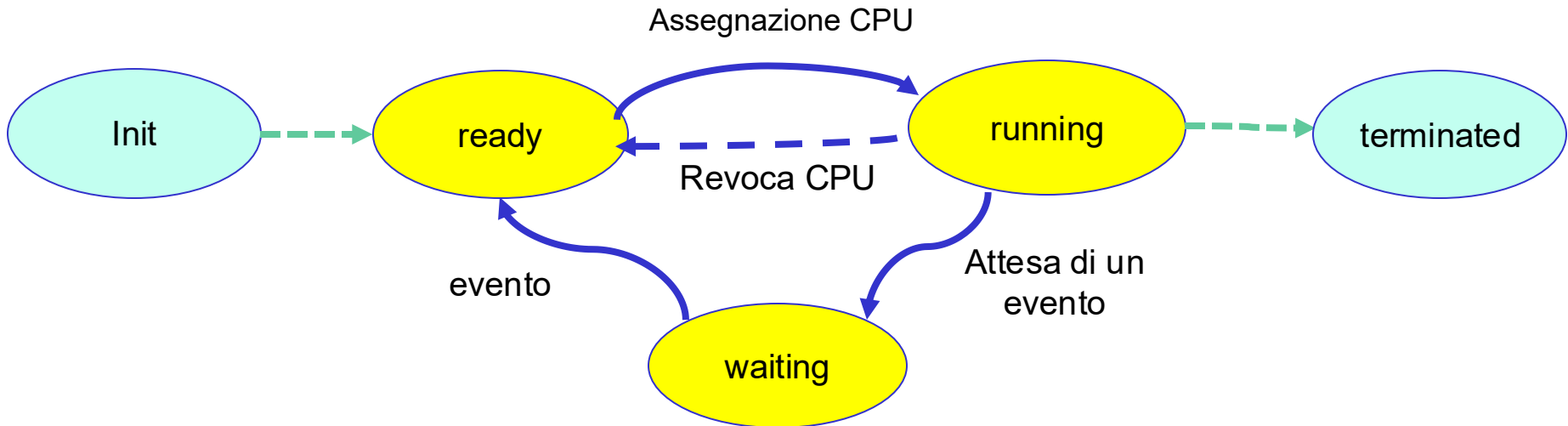
- è un sistema di elaborazione nel quale possono coesistere più attività (o **processi**) iniziate e non ancora terminate.
- Il sistema è in grado di gestire tali attività, suddividendo le risorse computazionali tra di esse, in modo tale che l'utente abbia l'illusione che l'esecuzione di ogni processo avvenga contemporaneamente all'esecuzione degli altri.

- **Programma concorrente:**

- programma la cui esecuzione determina la creazione di processi concorrenti.
- I processi concorrenti possono eventualmente interagire, per scambiarsi informazioni, oppure per sincronizzarsi nell'uso di risorse comuni.

Concorrenza

- La concorrenza è un'**astrazione** che viene implementata dal **sistema operativo** sfruttando l'hardware sottostante.
- **Sistema operativo:** il supporto è essenzialmente fornito dalla caratteristica di multiprogrammazione (multitasking/multithreading):
 - Lo scheduler può assegnare CPU a un processo, mentre altri processi non ancora terminati sono in attesa di un evento o nello stato di pronto



Concorrenza

Hardware: mette a disposizione un insieme di unità di elaborazione, insieme alla memoria e ai dispositivi di I/O.

In generale:

- nei sistemi concorrenti **non c'è relazione tra il numero di processi** che il sistema può gestire **ed il numero di CPU** disponibili a livello HW
- Spesso:

numero di processi >> numero di CPU

Parallelismo

- **Sistema Parallelo:** è un sistema di elaborazione nel quale è presente una moltitudine di unità di elaborazione (nodi), che vengono gestiti con le finalità:
 - **massimizzare** la velocità di calcolo (riduzione del tempo di esecuzione)
 - sfruttare in modo **efficiente** le risorse hardware disponibili
- **Programma parallelo:** è un'applicazione la cui esecuzione determina la creazione di un certo numero di processi, ognuno dei quali verrà eseguito su un'unità di elaborazione HW dedicata.
 - 👉 Ogni processo esegue in effettiva contemporaneità con gli altri processi (**reale parallelismo**).
 - 👉 Il programmatore, utilizzando opportuni linguaggi, può sfruttare il reale parallelismo tra processi per mettere in atto strategie volte a minimizzare i tempi di esecuzione (es: località dei dati, limitazione delle interazioni ecc.).

Programmi paralleli

A differenza dei sistemi concorrenti, nei sistemi paralleli **ad ogni processo** viene allocata **1 unità di elaborazione** in modo **esclusivo**.

Pertanto in un sistema parallelo:

$$\text{numero di processi} \leq \text{numero di CPU}$$

A seconda dell'algoritmo eseguito, i processi possono essere **indipendenti**, o **non**:

- **Processi non indipendenti** devono interagire: necessità di un opportuno supporto HW/SW all'interazione:
 - **Rete di comunicazione tra nodi**, requisito fondamentale: **efficienza**
 - **Linguaggio di programmazione** con costrutti specifici per esprimere l'interazione, con riferimento anche ad aspetti di legati all'HW (nodi, memoria, ecc.).

Uso di sistemi paralleli HPC

I sistemi paralleli, quando composti di un numero elevato di nodi, vengono tipicamente usati^{per} la soluzione di problemi computazionalmente intensivi in tempi contenuti. Pertanto, sistemi di questo tipo vengono chiamati anche **High Performance Computing systems (HPC)**.

Ad esempio, tipici ambiti applicativi HPC:

Ambito

Ricerca scientifica / Scienze fondamentali

Clima, meteorologia e scienze della terra

Scienze della vita e biotecnologie

Ingegneria, simulazione & progettazione industriale

Finanza / economia computazionale

Media, grafica

Intelligenza Artificiale e Machine/Deep Learning

Big Data / Data Analytics

Esempi / casi concreti

Fisica computazionale, astrofisica, cosmologia: simulazioni di dinamica stellare, formazione di galassie, modelli di collisione di particelle.

Previsioni meteorologiche, modellazione del clima, simulazioni di cambiamenti globali, modelli degli oceani, dell'atmosfera, attività vulcaniche, sismi.

Sequenziamento del DNA/genoma, biologia computazionale, modelli di proteine, scoperta di farmaci.

Analisi fluidodinamiche (CFD), simulazioni strutturali, simulazioni termo-meccaniche, virtual prototyping, ottimizzazione di componenti, simulazioni nel settore aerospaziale e automobilistico.

Analisi del rischio, modelli quantitativi, algoritmi di trading .

Rendering di effetti speciali, animazioni 3D, elaborazione video, realtà virtuale e aumentata.

Addestramento di modelli grandi su dataset estesi, ottimizzazione di algoritmi, inferenza su larga scala.

Analisi di grandi dataset, elaborazione post-osservazioni satellitari, gestione dati da osservazioni della Terra, monitoraggio ambientale, workflows di data science.

Relazione tra concorrenza e parallelismo

- **Concorrenza** come astrazione, realizzata dal sistema operativo e indipendente dalle caratteristiche dell'HW. Ogni applicazione concorrente è completamente astratta rispetto alle caratteristiche dell'HW utilizzato.
- **Parallelismo** come implementazione della concorrenza: sfrutta direttamente le risorse hardware multiple, con l'obiettivo della massima velocità di esecuzione.

👉 Un'applicazione concorrente può essere eseguita in parallelo

👉 Un'applicazione parallela non è necessariamente concorrente, perché la sua struttura, in generale, dipende dall'hardware (es. nodi, memoria, ecc.)

Modelli di interazione tra processi: richiami

Memoria comune (Shared Memory):

- Processo= **thread**
- **condivisione** spazio indirizzamento → richiede **sincronizzazione**

Vantaggi: efficienza

Svantaggi: complessità nella gestione

Scambio di messaggi (Message Passing):

- Processo=**task**
- **non c'è condivisione** di dati → comunicazione/sincronizzazione

Vantaggi: isolamento processi, minore complessità

Svantaggi: overhead meccanismi comunicazione

Modelli di interazione

Sistemi Concorrenti:

La concorrenza viene realizzata dal kernel del sistema operativo, che fornisce supporto ai processi e alla loro interazione, secondo il modello adottato nel SO (multithreading o multitasking).

La **scelta del modello** di interazione dipende quindi dal **modello adottato nel SO**:

- **Multithreading** → shared memory
- **Multitasking** → message passing

Sistemi Paralleli:

Parallelismo come implementazione della concorrenza: sfrutta direttamente le risorse hardware multiple, con l'obiettivo della massima velocità di esecuzione.

La **scelta del modello** di interazione dipende anche **dalle caratteristiche dell'architettura HW**:

- Se la memoria è condivisa → modello shared memory
- Se la memoria è distribuita tra i nodi → modello a scambio di messaggi