

ALGORITMI PARALLELI e DISTRIBUITI

DOCENTE: PALANO BEATRICE

homepage: palano.di.unimi.it

email: palano@di.unimi.it

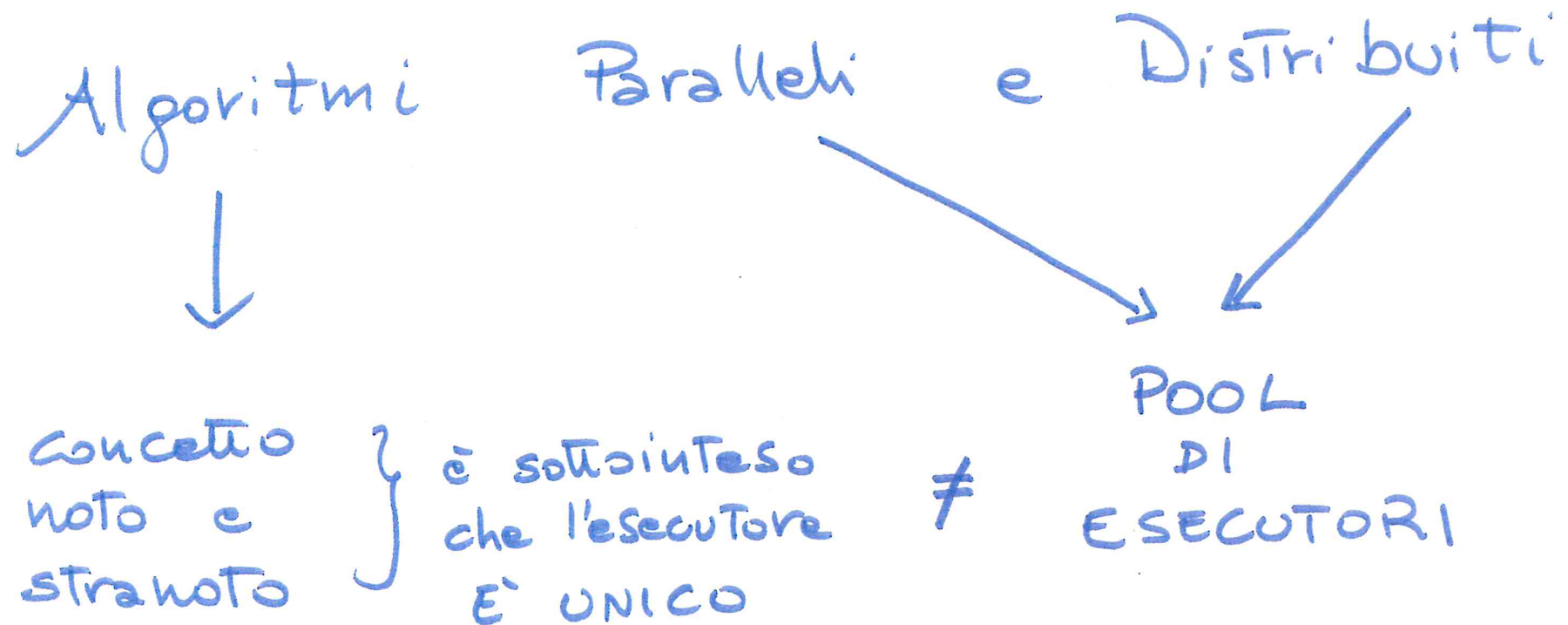
ricevimento: su appuntamento

4011 IV piano via Celoria 18

PROGRAMMA DEL CORSO :

di cosa parleremo ?

Partiamo dal titolo del corso :



Problematiche degli algoritmi sequenziali:

- progettare : Tecniche come ad esempio
divide et impera,
programmazione dinamica,
greedy, ...
- valutare le prestazioni : Tempo, spazio di
memoria
- codificare : implementare con un opportuno
linguaggio di programmazione

Analogamente le stesse problematiche
possono essere affrontate per gli
algoritmi paralleli e distribuiti

Gli Algo. paral. e distr. sono risolti ad un
POOL DI ESECUTORI

Vi sono due situazioni ben diverse
ispirate a casi reali



I CASO

"Una squadra
in cui batte
un solo cuore"

II CASO

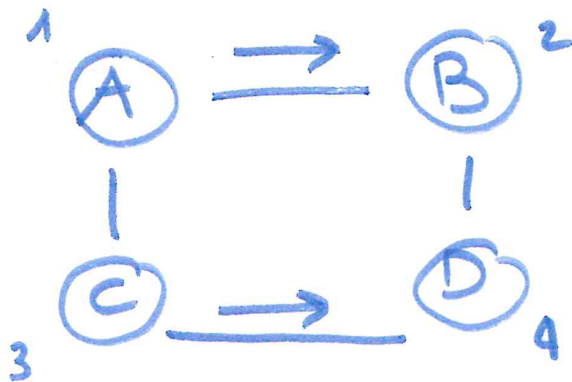
"ogni membro del
pool
è un mondo a parte"

I CASO: pool di esecutori tali che:

- vanno all'unisono, ovvero abbiamo un clock centrale
- possono condividere risorse (es. memoria)
o ammettere opportuni collegamenti tra loro

Esempio:

Sommare 4 numeri \Rightarrow



Numeri: A, B, C, D

processori: 1, 2, 3, 4

Programma

send (1, 2), send (3, 4)
A + B, C + D

send (2, 4)
A + B + C + D

QUANTI COLPI DI CLOCK?

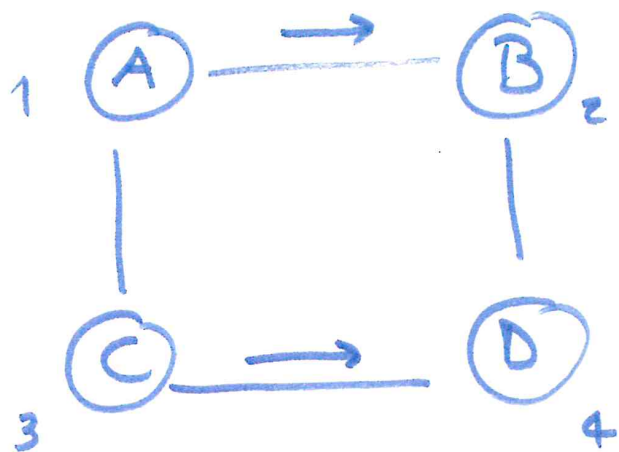
Risp. 4

II CASO : pool di esecutori tali che :

- ognuno ha un proprio clock centrale (i propri tempi)
- sono collegati Tra loro (non possiamo assumere memoria in comune)

Nota bene : non possiamo assumere alcun Tipo di Sincronizzazione gratuita,
DEVONO scambiarsi messaggi !!

Esempio : Sommare 4 numeri



- send (1,2) e send (3,4)
- potrebbero avvenire in tempi \neq
- per fare la somma Tot. il proc. 4 potrebbe aspettare
- potrebbero servire segnali di coordinamento

Teoria e realtà: MOTIVAZIONI

Nascita di architetture parallele e distribuite:
seconda metà degli anni '60

di conseguenza si è sviluppata la Teoria

④ Esempi di architetture parallele:

- supercomputer: cluster di processori con incredibili prestazioni di calcolo (CDC, CRAY, ...)

Il più potente nel 2018: SUMMIT IBM
200 petaFlops = 10^{15} operazioni x 200
Floating Points

al secondo

Scopo: Simulazioni di sistemi complessi
(fisici, economici, militari, ...)

- GPU : sarebbero dedicate alla grafica ma vengono utilizzate in senso più generale
- Multicore processor : attuali processori con più core ottimizzano l'assorbimento di energia e relativi problemi di raffreddamento
- Circuiti Integrati : componenti di sistemi di calcolo insieme di gate opportunamente connessi

② Esempi di architetture distribuite:

- reti di calcolatori : Internet (dagli anni '60)
connettono dispositivi molto diversi
protocollo di comunicazione TCP/IP

- Reti mobili: la Topologia di connessione dei dispositivi cambia in continuazione

- Reti di sensori: dispositivi con limitata capacità tenuti in "stand by" per la maggior parte del tempo

scopo: monitoraggio di certe realtà
hanno bisogno di meccanismi di
"wake-up", "acknowledge", "recovery", ...

Nel parallelo il fattore rilevante è IL TEMPO

Es. Nel sommare 4 numeri i passi sono 4,
ma nel sommare 1000 i passi sono solo 10!!

Nel distribuito il fattore rilevante è IL COORDINAMENTO

Meno messaggi = più veloce!!

PROBLEMATICHE NEL CASO PARALLELO E DISTRIBUITO

- progettare : nuove idee
- valutare le prestazioni : nuove misure

NUOVE IDEE, perché?

- Un algoritmo sequenziale efficiente non sempre porta ad un algoritmo parallelo e distribuito efficiente
- Buoni algoritmi paralleli non sempre portano a buoni algoritmi distribuiti e viceversa

"Nel distribuito dobbiamo gestire problemi di ritardo della comunicazione"

AVVERTENZA : Per le misure:
Non faremo riferimento ad alcuna architettura particolare !!

PROGRAMMA DEL CORSO

Per ogni paradigma

- 1 MODELLO TEORICO
- 2 COME VALUTARE LE PERFORMANCE DEGLI ALGORITMI
- 3 SEMPLICI PROBLEMI PER APPRENDERE LE TECNICHE

① CASO PARALLELO

- A) 1 PRAM (comunicazione immediata) : memoria condivisa
- 2 risorse di calcolo : Tempo, hardware = n^2 processori
 - 3 problemi :
 - Sommatoria
 - somme prefisse
 - Ordinamento

B) 1 modelli a memoria distribuita :

- ARRAY LINEARI
- MESH

2 risorse di calcolo : Tempo, hardware



(n.b.) processori ~~XXXXXX~~ ~~XXXXXX~~ non connessi direttamente
comunicano più lentamente

3 problemi : - shuffle
- max
- ordinamento

② CASO DISTRIBUITO

1 def. del modello astratto

2 risorse di calcolo : Tempo, n° di msg = congestione

3 problemi : - broadcast
- wake up
- traversal
- spanning Tree
- election
- routing