

Per ottenere le credenziali di accesso al cluster,
inviare IMMEDIATAMENTE a

valeria.mele@unina.it

un messaggio con subject:

PDC-<nome cognome matricola> credenziali

contenente, oltre a nome cognome matricola (da
ripetere nel testo del messaggio) il

CODICE FISCALE!

a. a. 2022/2023 G. Laccetti - Parallel and Distributed Computing

1

Parallel and Distributed Computing

► Gruppo Teams

<https://teams.microsoft.com/l/team/19%3aFYdihBDwupy2Lsk94yRlpJxXbKy03094zJ0Qz8HC8JY1%40thread.tacv2/conversations?groupId=1ddd353f-7a8e-4cd6-a112-d4d3967eb8a5&tenantId=2fcfe26a-bb62-46b0-b1e3-28f9da0c45fd>

► Sito web docenti

www.docenti.unina.it

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

2

Parallel and Distributed Computing

a.a. 2023-2024

Prof. Giuliano Laccetti

Università degli Studi di Napoli Federico II

3

slides basate sul testo
A. Murli «Lezioni di Calcolo Parallelo»
e
su appunti di lezioni di Calcolo Parallelo e Distribuito tenute dal prof. A. Murli

a. a. 2022/2023 G. Laccetti – Parallel and Distributed Computing

4

CORSO

- ⇒ Lezioni ed esercitazioni in aula
- ⇒ Esercitazioni in laboratorio

ESAMI

- ⇒ Prova scritta in aula +
elaborati consegnati durante il corso / prova pratica in laboratorio
prova orale

Modalità

a. a. 2022/2023 G. Laccetti - Parallel and Distributed Computing

5

Riferimenti Bibliografici – 1



Lezioni di Calcolo Parallelo

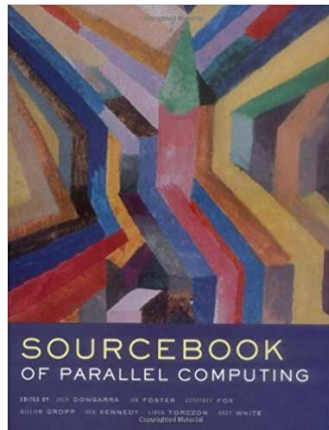
Almerico Murli

Liguori Editore

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

6

Riferimenti Bibliografici – 2



Sourcebook of Parallel Computing

Jack Dongarra, Ian Foster, Geoffrey C. Fox, William Gropp, Ken Kennedy, Linda Torczon, Andy White

Morgan Kaufmann Edition

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

7

Introduzione al corso

»» Perché Calcolo Parallelo?

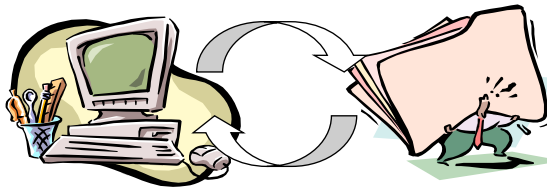
a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

8

Obiettivo del corso

Fornire idee, strumenti software e metodologie
alla base della
risoluzione computazionale di un problema
mediante
calcolatore parallelo

Attività di
laboratorio



Lezioni
in aula

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

9

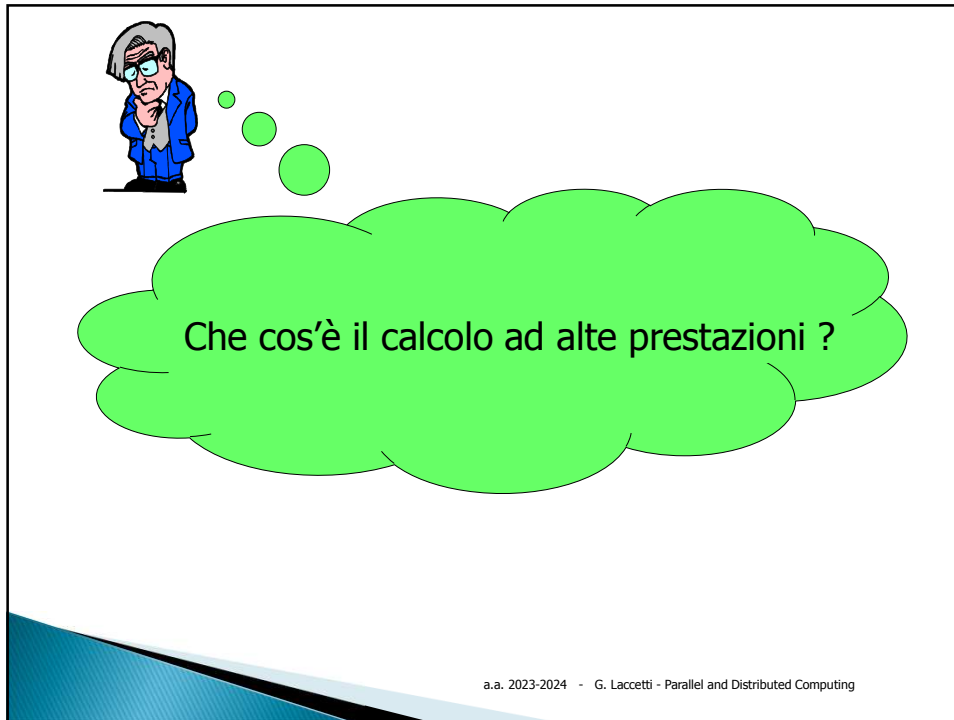
L'attività di laboratorio

Progetto, sviluppo, analisi e implementazione
di algoritmi in ambienti ad
alte prestazioni



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

10



11

Supercomputing

The real purpose of having a **super** computer is to solve problems

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

12

Supercomputing

The real purpose of having a
super computer
is to solve problems

Supercomputing is
To solve a problem by using a supercomputer

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

13

Supercomputing

Il termine “**supercalcolatore**” si riferisce ad un
sistema che fornisce le prestazioni più elevate
(in quel dato momento).

(’80s)

La prestazione è misurata dal tempo necessario per risolvere una
particolare applicazione
(**application-dependent**)

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

14

Supercomputing

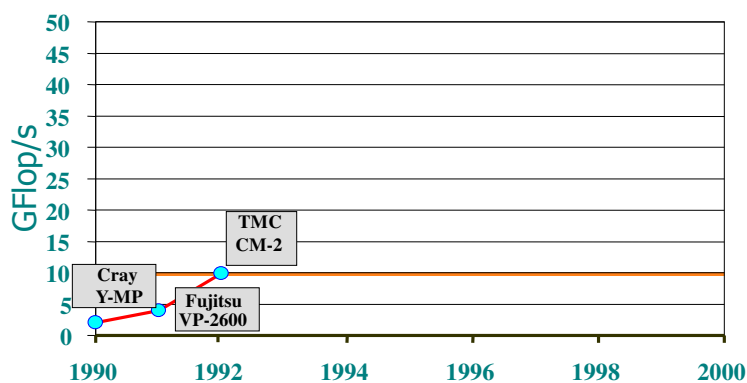
Dal 1993

- ▶ **TOP 500**: lista dei calcolatori più veloci nel mondo
(ideata tra gli altri da Jack Dongarra, recentissimo vincitore del Turing Award)
- ▶ **Rmax**: performance misurata dai benchmark di LINPACK per la risoluzione di $AX=B$
- ▶ **Rpeak**: massima performance raggiungibile („in teoria“) in base alle caratteristiche dell'hardware

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

15

I calcolatori più veloci

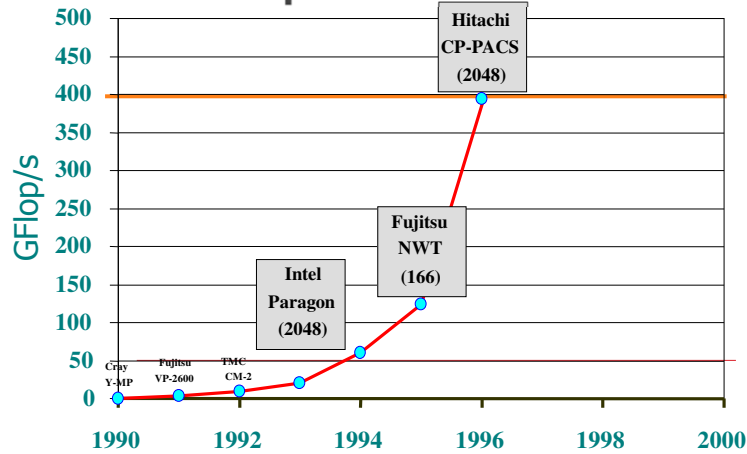


**Un calcolo che nel 1980 durava 1 anno (CRAY 1, 100 Mflop)
può essere effettuato nel 1992 (CM-2, 10 Gflop)
in circa 87 ore**

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

16

I calcolatori più veloci

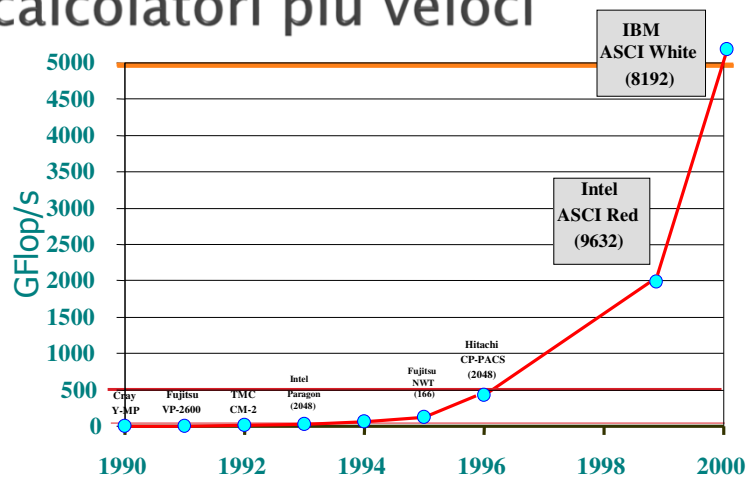


Un calcolo che nel 1980 durava 1 anno (CRAY 1, 100 Mflop) può essere effettuato nel 1996 (Hitachi, 400 GFlop) in circa 2 ore!

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

17

I calcolatori più veloci

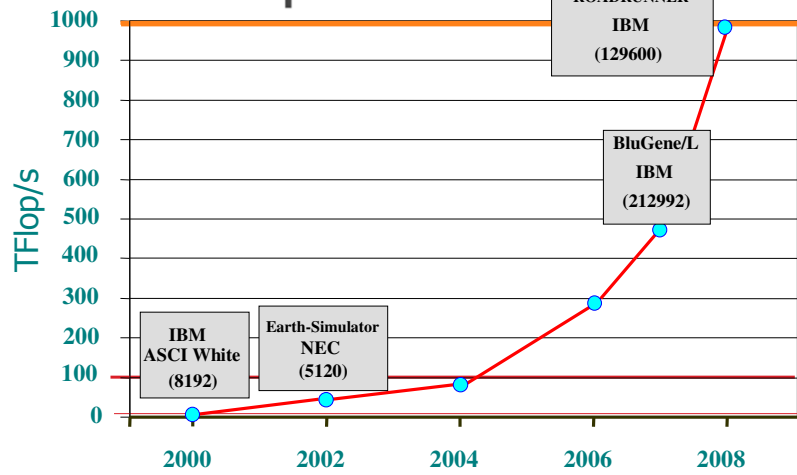


Un calcolo che nel 1980 durava 1 anno (CRAY 1, 100 Mflop) può essere effettuato nel 2000 (ASCI White, 7 Tflop) in circa 7 minuti!

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

18

I calcolatori più veloci

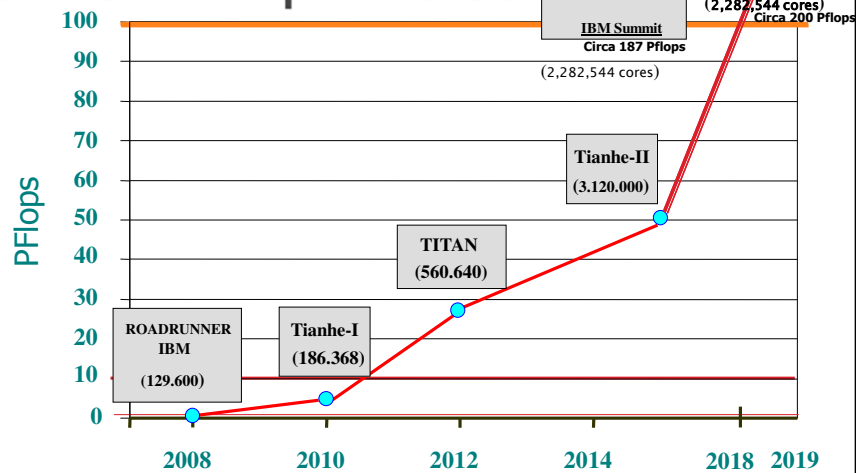


Un calcolo che nel 1980 durava 1 anno (CRAY 1, 100 Mflop) può essere effettuato nel 2008 (ROADRUNNER, 1 Pflop) in meno di 3 secondi!

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

19

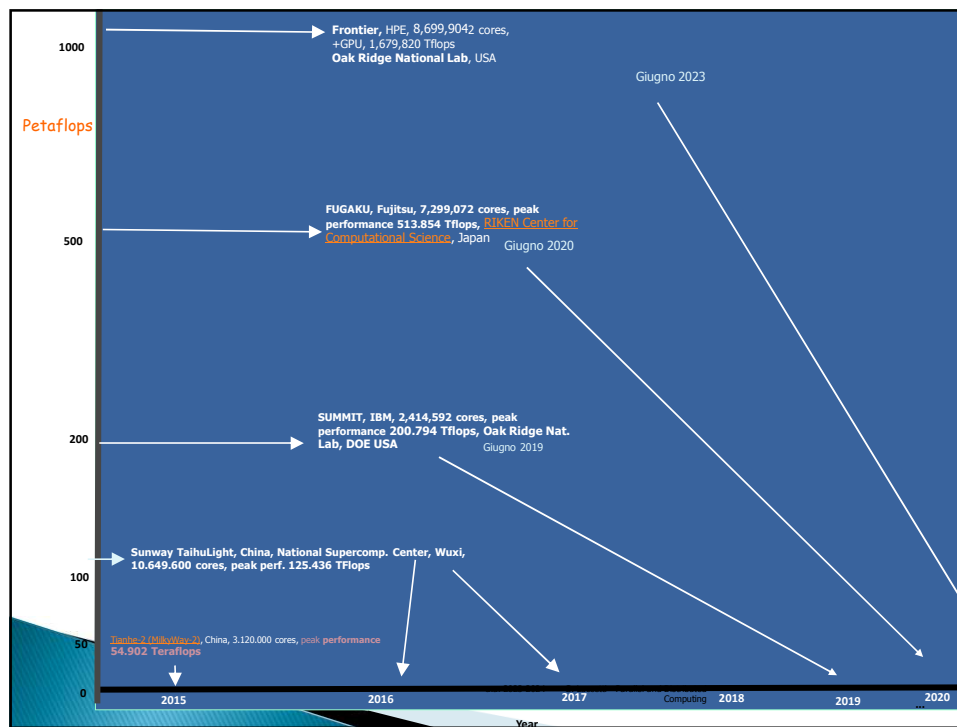
I calcolatori più veloci



Un calcolo che nel 1980 durava 1 anno (CRAY 1, 100 Mflops) può essere effettuato nel 2019 (IBM Summit, Oak Ridge Lab, 200 PFlops) in circa 15 millesimi di secondo!

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

20



21

- ▶ problema che richiede 3.15×10^{15} flop
- ▶ tempo di risoluzione su di un CRAY 1 (1980) con prestazioni di 100 Mflops = 1 anno ($\sim 3.15 \times 10^7$ secs)

22

Un calcolo che nel 1980 durava 1 anno (CRAY 1, 100 Mflops) ,

Può essere effettuato **ora (giugno 2023)**

(Frontier, HPE, Oak Ridge National Lab, USA **1.680 Eflops di picco**)

in **MENO di 2 millesimi di secondo!!**

(1,88 millesimi per la precisione)

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

23

23

Un calcolatore parallelo e'...



Frontier – HPE (Oak Ridge National Lab, USA)

Processore:
Optimized 3rd Generation EPYC 64C @2.0 GHz

Performance sostenuta: 1,194 PFlops

Picco di performance: **1,680 PFlops**

Notazione USA: la virgola per migliaia ecc .. il punto per i decimali
1,680 Pflops sono 1.680 Eflops !!!

Calcolatore Parallelo

1 miliardo di miliardi di operazioni al secondo!



1 Tflops = 10^{12} flop/sec

1 Pflops = 10^{15} flop/sec

1 Eflops = 10^{18} flop/sec

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

24


24

| | Cost. | computer | sito | Ncores | Linpack bench. (Pflops) |
|------|------------------|---------------------------|--|--|----------------------------|
| 1 | HPE | Frontier | ORNL (Oak Ridge National Lab) USA | 8,699,904 | 1,194.00 |
| 2 | Fujitsu | Fugaku | Riken Center for Computational Science JAPAN | 7,630,848 | 442.010 |
| 3 | HPE | LUMI | EuroHPC JU European Union, location: Kajaani, FINLAND. | 2,220,288 | 309.100 |
| 4 | BULL- Sequana | Leonardo | EuroHPC/CINECA Bologna, Italy | 2,414,592 | 148.60 |
| 15 | DELL | HPC5 | Exploration & Production ENI (Milano) ITALY | 699,760 (Xeon Gold 6252 24C 2.1GHz) | 35.450 |
| | | | | | |
| 2005 | NEC (2005) | Earth Simulator (2005) | Yokohama Giappone (2005) | 5,120 proc. NEC SX/6 (2005) | circa 0.036 (2005) |

Supercomputing

Calcolo ad "alte prestazioni"

- Anni '70 - '80: 10^6 flops (MFLOPs)
 - ✓ Calcolatori sequenziali-scalari (CDC 7600, IBM 360)
- Anni '81 - '90: 10^9 flops (GFLOPs)
 - ✓ Calcolatori vettoriali (CRAY 1, CRAY X-MP)
- Anni '91 - 2000: 10^{12} flops (TFLOPs)
 - ✓ Calcolatori a parallelismo massiccio (CRAY T3D, ASCI White)
- Anni 2001 - 2010: 10^{15} flops (PFLOPs)
 - ✓ Calcolatori a parallelismo massiccio (Roadrunner)
- Anni 2011 - 2020: 10^{17} flops (553 PFLOPs - Fugaku, Japan)
 - ✓ Calcolatori a parallelismo massiccio (Tianhe-2; Sunway Tiahu, Summit)
- Anni 2021-20xx: **10^{18} flops (EFLOPs - Frontier, HPE, Oak Ridge)**
 - 1.6 Eflops Rpeak, 1.1 Eflops Rmax)**



Ma ... il calcolo ad alte prestazioni
è davvero necessario?

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

27



Una Ferrari o
una Panda?

dipende dall'uso!



In città':
traffico, parcheggi,
consumo...

In autostrada: **velocita',**
affidabilità, sicurezza,....



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

28

Difesa del territorio?

Nella Guerra del Golfo, il Calcolo ad Alte Prestazioni, ha guidato Patriot contro gli Scud Iracheni

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

Difesa del territorio?

Nella Guerra del Golfo, il Calcolo ad Alte Prestazioni, ha guidato Patriot contro gli Scud Iracheni

Il missile deve essere distrutto prima che arrivi sull'obiettivo

Entro 2 minuti dal lancio del missile nemico!

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

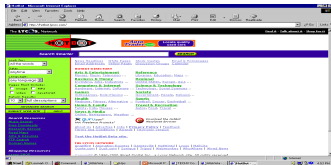
Altre applicazioni

altavista: SEARCH



► Ricerca su Internet

- Ogni giorno circa un milione di persone interroga su Internet un motore di ricerca



Google™



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

31

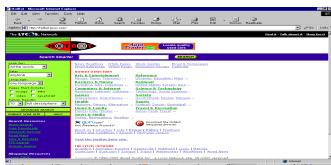
Altre applicazioni

altavista: SEARCH



► Ricerca su Internet

- Ogni giorno circa milioni e milioni di persone interrogano su Internet un motore di ricerca
 - Quanto si è disposti ad aspettare?



Google™



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

32

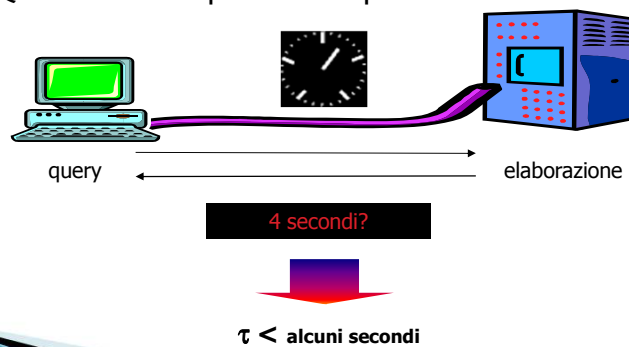
Altre applicazioni

alta**vista**: SEARCH



► Ricerca su Internet

- Ogni giorno circa un milione di persone interroga su Internet un motore di ricerca
- Quanto si è disposti ad aspettare?

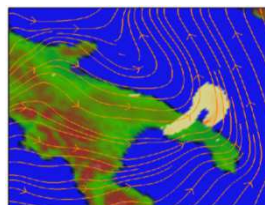
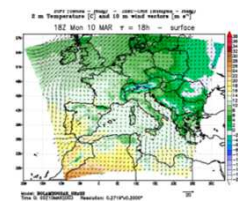


a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

33

Altre applicazioni

► Previsioni metereologiche



- superficie da monitorare
20 milioni di Km²
- altezza sul livello del mare
20 Km
- discretizzazione dello spazio 3D
mediante cubi di lato 100 m
in 1 km³: 10 x 10 x 10 = 10³ cubi

$$20 \times 10^6 \text{ km}^2 \times 20 \text{ km} = 4 \times 10^8 \text{ km}^3$$

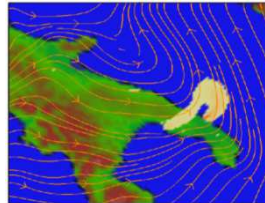
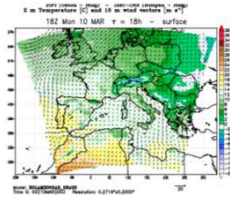
$$4 \times 10^8 \times 10^3 = 4 \times 10^{11} \text{ cubi}$$

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

34

Altre applicazioni

► Previsioni metereologiche



- superficie da monitorare
20 milioni di Km²
- altezza sul livello del mare
20 Km
- discretizzazione dello spazio 3D
mediante cubi di lato 100 m
in 1 km³:10 x 10 x 10 = 10³ cubi

$$20 \times 10^6 \text{ km}^2 \times 20 \text{ km} = 4 \times 10^8 \text{ km}^3$$

$$4 \times 10^8 \times 10^3 = 4 \times 10^{11} \text{ cubi}$$

Il modello deve fornire le previsioni per i prossimi 2 giorni.

Supponiamo che l'algoritmo effettui, per ogni ora di simulazione, 10² flop per cubo

► per l'intera superficie sono richiesti $4 \times 10^{11} \times 10^2 \text{ flop}$

► Per **2 giorni**: $4 \times 10^{11} \times 10^2 \times 48 \text{ flop}$

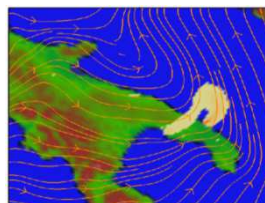
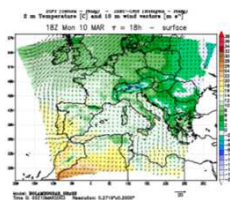
2 x 10¹⁵ operazioni

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

35

Altre applicazioni

► Previsioni metereologiche



- superficie da monitorare
20 milioni di Km²
- altezza sul livello del mare
20 Km
- discretizzazione dello spazio 3D
mediante cubi di lato 100 m
in 1 km³:10 x 10 x 10 = 10³ cubi

$$20 \times 10^6 \text{ km}^2 \times 20 \text{ km} = 4 \times 10^8 \text{ km}^3$$

$$4 \times 10^8 \times 10^3 = 4 \times 10^{11} \text{ cubi}$$

Il modello deve fornire le previsioni per i prossimi 2 giorni.

Supponiamo che l'algoritmo effettui, per ogni ora di simulazione, 10² flop per cubo

► per l'intera superficie sono richiesti $4 \times 10^{11} \times 10^2 \text{ flop}$

► Per **2 giorni**: $4 \times 10^{11} \times 10^2 \times 48 \text{ flop}$

2 x 10¹⁵ operazioni

Con un PC

1 x 10¹⁰ flop/s

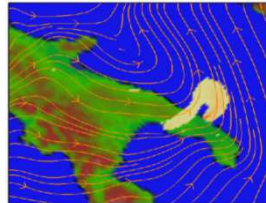
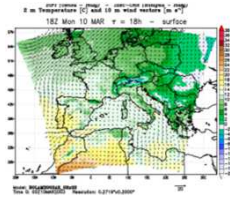
= 2 x 10⁵ sec = **2 giorni**

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

36

Altre applicazioni

► Previsioni metereologiche



- superficie da monitorare
20 milioni di Km²
- altezza sul livello del mare
20 Km
- discretizzazione dello spazio 3D
mediante cubi di lato 100 m
in 1 km³:10 x 10 x 10 = 103 cubi

$$20 \times 10^6 \text{ km}^2 \times 20 \text{ km} = 4 \times 10^8 \text{ km}^3$$

$$4 \times 10^8 \times 10^3 = 4 \times 10^{11} \text{ cubi}$$

Il modello deve fornire le previsioni per i prossimi 2 giorni.

Supponiamo che l'algoritmo effettui, per ogni ora di simulazione, 10^2 flop per cubo

► per l'intera superficie sono richiesti $4 \times 10^{11} \times 10^2$ flop

► Per **2 giorni**: $4 \times 10^{11} \times 10^2 \times 48$ flop

2 x 10¹⁵ operazioni

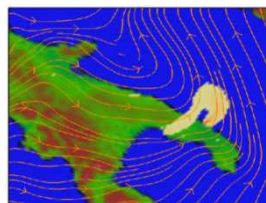
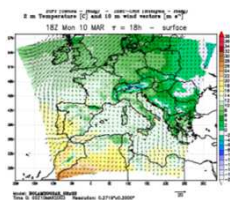
30 minuti

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

37

Altre applicazioni

► Previsioni metereologiche



- superficie da monitorare
20 milioni di Km²
- altezza sul livello del mare
20 Km
- discretizzazione dello spazio 3D
mediante cubi di lato 100 m
in 1 km³:10 x 10 x 10 = 103 cubi

$$20 \times 10^6 \text{ km}^2 \times 20 \text{ km} = 4 \times 10^8 \text{ km}^3$$

$$4 \times 10^8 \times 10^3 = 4 \times 10^{11} \text{ cubi}$$

Il modello deve fornire le previsioni per i prossimi 2 giorni.

Supponiamo che l'algoritmo effettui, per ogni ora di simulazione, 10^2 flop per cubo

► per l'intera superficie sono richiesti $4 \times 10^{11} \times 10^2$ flop

► Per **2 giorni**: $4 \times 10^{11} \times 10^2 \times 48$ flop

2 x 10¹⁵ operazioni

E' necessario 1 TFLOPs !!!!!

1 x 10¹² flop/s

= 1.8 x 10³ sec = 30 minuti

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

38

Come ridurre i tempi di risposta (*turnaround time*) di una simulazione?

► In generale

- una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

$T(N)$ = complessità di tempo dell'algoritmo
 Dipendenza dall'algoritmo

μ = tempo di esecuzione di 1 op. f.p.
 Dipendenza dal calcolatore

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

39

Soluzione A

► In generale

- una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

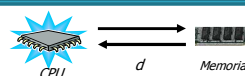
$$\tau = k \cdot T(n) \cdot \mu$$

μ = tempo di esecuzione di 1 op. f.p.
 Dipendenza dal calcolatore

Si può ridurre migliorando la tecnologia

Velocità della luce $c = 3 \times 10^8$ m/s

$d = \mu \cdot c$



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

40

Soluzione A

► In generale

- una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

$T(N)$ = complessità di tempo dell'algoritmo

Dipendenza dall'algoritmo

μ = tempo di esecuzione di 1 op. f.p.

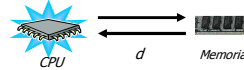
Dipendenza dal calcolatore

Si può ridurre migliorando la tecnologia

Limiti tecnologici
(packaging e raffreddamento)

Velocità della luce $c = 3 \times 10^8$ m/s

$$d = \mu \cdot c$$



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

41

Soluzione A

► In generale

- una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

μ = tempo di esecuzione di 1 op. f.p.

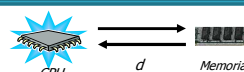
Dipendenza dal calcolatore

Si può ridurre migliorando la tecnologia

2002
STOP!!!

Velocità della luce $c = 3 \times 10^8$ m/s

$$d = \mu \cdot c$$



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

42

Soluzione B

- In generale
 - una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

$T(N)$ = complessità di tempo dell'algoritmo

Dipendenza dall'algoritmo

Si può ridurre riorganizzando l'algoritmo

E' possibile dimostrare
(teoria della complessità degli algoritmi)
che per alcune classi di problemi
esistono algoritmi con complessità di
tempo minima
(algoritmi ottimali)

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

43

Soluzione B

- In generale
 - una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

$T(N)$ = complessità di tempo dell'algoritmo

Dipendenza dall'algoritmo

Si può ridurre riorganizzando l'algoritmo

Progettare un algoritmo 10x più
veloce equivale a costruire un
calcolatore 10x volte più potente

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

44

Soluzione B

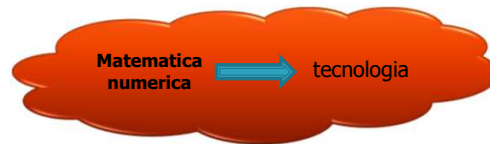
- In generale
 - una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

$T(N)$ = complessità di tempo dell'algoritmo

Dipendenza dall'algoritmo

Si può ridurre riorganizzando l'algoritmo



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

45

Soluzione C: CALCOLO PARALLELO

- In generale
 - una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

Decomporre un problema di dimensione N
in P sottoproblemi di dimensione N/P
e risolverli contemporaneamente su più calcolatori

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

46

Soluzione C: CALCOLO PARALLELO

- In generale
 - una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

Tempo d'esecuzione per un problema di dimensione n:
T(n)

Decomporre un problema di dimensione N
in P sottoproblemi di dimensione N/P
e risolverli contemporaneamente su più calcolatori



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

47

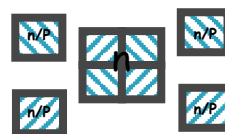
Soluzione C: CALCOLO PARALLELO

- In generale
 - una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

Tempo d'esecuzione per un problema di dimensione n/P:
T(n/P)

Decomporre un problema di dimensione N
in P sottoproblemi di dimensione N/P
e risolverli contemporaneamente su più calcolatori



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

48

Soluzione C: CALCOLO PARALLELO

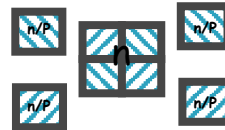
- In generale
 - una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

La quantità che meglio esprime l'importanza della decomposizione del problema è lo **SCALE UP**

$$\frac{T(n)}{P \cdot T\left(\frac{n}{P}\right)}$$

Decomporre un problema di dimensione N
in P sottoproblemi di dimensione N/P
e risolverli contemporaneamente su più calcolatori



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

49

Soluzione C: CALCOLO PARALLELO

- In generale
 - una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

Decomporre un problema di dimensione N
in P sottoproblemi di dimensione N/P
e risolverli contemporaneamente su più calcolatori

CALCOLO PARALLELO

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

50

Soluzione C: CALCOLO PARALLELO

- ▶ In generale
 - una rappresentazione semplificata del tempo richiesto dalla risoluzione numerica di un problema è:

$$\tau = k \cdot T(n) \cdot \mu$$

Decomporre un problema di dimensione N
in P sottoproblemi di dimensione N/P
e risolverli contemporaneamente su più calcolatori

Sviluppo di nuovi strumenti
computazionali
Hardware/software/algoritmi

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

51

Introduzione al corso

»» Tipi di parallelismo

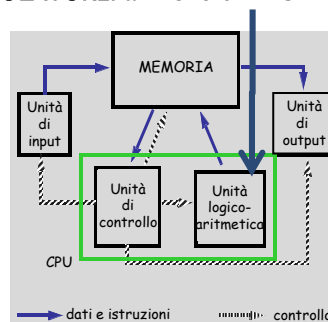
a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

52

Tipi di parallelismo

Parallelismo Temporale

- ▶ Tecnica della catena di montaggio (pipeline)
- ▶ I tre operai eseguono contemporaneamente fasi successive dello stesso lavoro
- ▶ In un **CALCOLATORE**: a livello di ALU



**Parallelismo
on-chip**

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

53

Unità logico aritmetica

somma di 100 numeri floating point

$$c_i = a_i + b_i \quad i=1,100$$

L'unità funzionale per l'addizione floating point è divisa in segmenti

Ciascun segmento è preposto alla esecuzione di una fase dell'operazione, ad esempio per la somma:

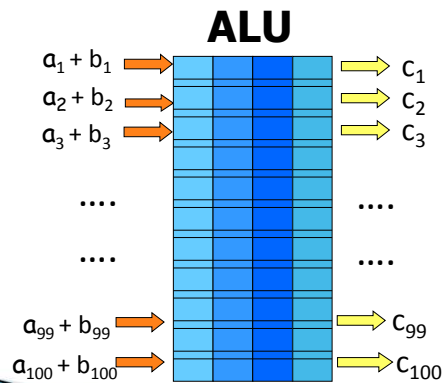
| | |
|---------------------------|--------|
| CONFRONTO DEGLI ESPONENTI | FASE 1 |
| SHIFT DELLA MANTISSA | FASE 2 |
| SOMMA DELLE MANTISSE | FASE 3 |
| NORMALIZZAZIONE | FASE 4 |

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

54

Unità logico aritmetica

somma di 100 numeri floating point
 $c_i = a_i + b_i \quad i=1,...,100$



Unità tradizionale

$N=100$, 4 fasi
 t = tempo di una fase

$$T=N*4t$$

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

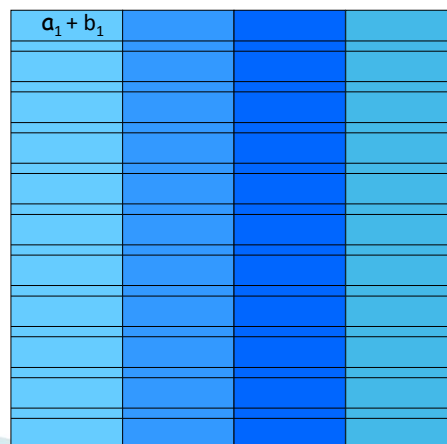
55

Unità logico aritmetica pipelined

somma di 100 numeri floating point
 $c_i = a_i + b_i \quad i=1,...,100$

$N=100$, 4 fasi
 t = tempo di una fase

T=



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

56

Unità logico aritmetica pipelined

somma di 100 numeri floating point

$$c_i = a_i + b_i \quad i=1,\dots,100$$

N=100, 4 fasi

t= tempo di una fase

T=

[illegible]

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

57

Unità logico aritmetica pipelined

somma di 100 numeri floating point

$$c_i = a_i + b_i \quad i=1,\dots,100$$

N=100, 4 fasi

t= tempo di una fase

T=

[illegible]

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

58

Unità logico aritmetica pipelined

somma di 100 numeri floating point
 $c_i = a_i + b_i \quad i=1,...100$

$N=100$, 4 fasi
 t = tempo di una fase

$T=$

| | | | |
|-------------|-------------|-------------|--|
| | | $a_1 + b_1$ | |
| | $a_2 + b_2$ | | |
| $a_3 + b_3$ | | | |
| $a_4 + b_4$ | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

59

Unità logico aritmetica pipelined

somma di 100 numeri floating point
 $c_i = a_i + b_i \quad i=1,...100$

$N=100$, 4 fasi
 t = tempo di una fase

$T=4t+$

| | | | |
|-------------|-------------|-------------|-------------|
| | | | $a_1 + b_1$ |
| | | $a_2 + b_2$ | |
| | $a_3 + b_3$ | | |
| $a_4 + b_4$ | | | |
| $a_5 + b_5$ | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

60

Unità logico aritmetica pipelined

somma di 100 numeri floating point

$$c_i = a_i + b_i \quad i=1,...,100$$

$N=100$, 4 fasi
 t = tempo di una fase

$$T=4t+(N-1)t$$

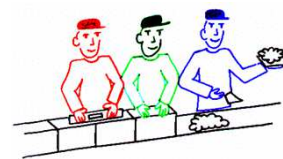


a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

61

Tipi di parallelismo

- ▶ **Parallelismo Temporale**
 - ▶ Tecnica della catena di montaggio (pipeline)
 - ▶ I tre operai eseguono contemporaneamente fasi successive dello stesso lavoro
- ▶ **In un CALCOLATORE:** a livello di ALU
- ▶ Architetture RISC, processori vettoriali



Il primo "supercomputer" della storia
 CRAY 1 (Los Alamos Laboratory, 1976)

133 Mflops
 8 registri scalari
 8 registri vettoriali



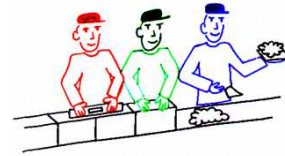
a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

62

Tipi di parallelismo

▶ Parallelismo Temporale

- ▶ Tecnica della catena di montaggio (pipeline)
- ▶ I tre operai eseguono contemporaneamente fasi successive dello stesso lavoro
- ▶ In un **CALCOLATORE**: a livello di ALU
- ▶ Architetture RISC, processori vettoriali



Attualmente
tutti i processori
utilizzano una struttura a pipeline per
migliorare le loro prestazioni.



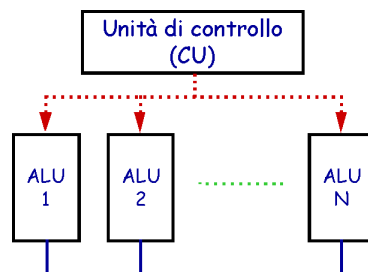
a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

63

Tipi di parallelismo

▶ Parallelismo Spaziale

- ▶ I tre operai eseguono contemporaneamente la stessa azione su mattoni diversi
- ▶ In un **CALCOLATORE**:
 - più ALU per la stessa CU



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

64

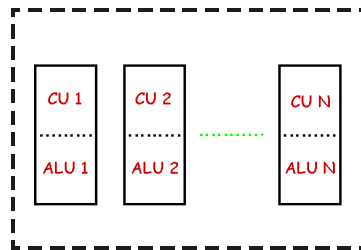
Tipi di parallelismo

▶ Parallelismo Asincrono

- ▶ I tre operai eseguono contemporaneamente azioni diverse su parti diverse

▶ In un CALCOLATORE:

- Più CPU
- Differenti processori cooperano eseguendo istruzioni diverse su dati diversi



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

65

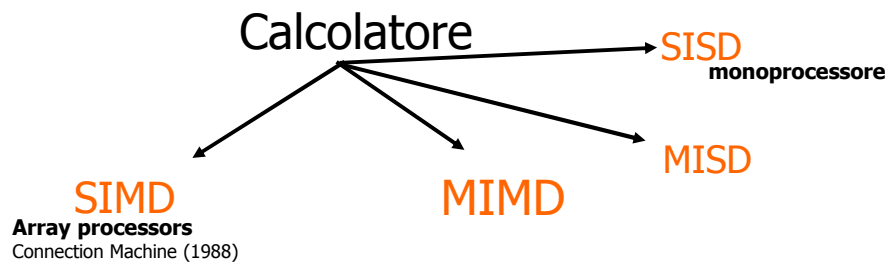
Introduzione al corso

»» Tassonomia di Flynn

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

66

Tassonomia di Flynn (dal 1965)

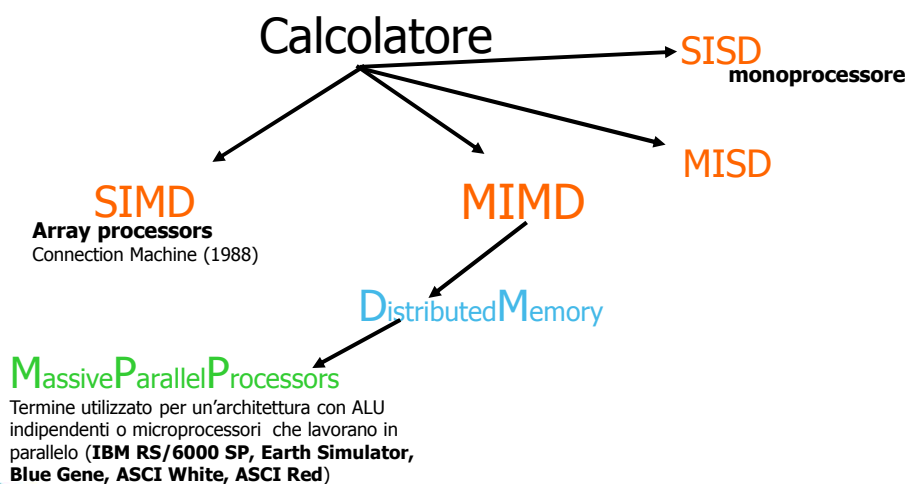


Negli anni '60 non erano ancora materialmente diffuse macchine diverse dai monoprocessore (SISD): all'inizio degli anni '70 erano nati alcuni array processors (che Flynn stesso definì come SIMD) e da alcune compagnie venivano le prime proposte di architetture che Flynn comprese nella sua definizione di macchina MIMD.

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

67

Tassonomia di Flynn (dal 1965)



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

68

Esempio MIMD-MPP

A memoria distribuita

MPP

Massive Parallel Processor:
processori strettamente integrati,
che danno un'immagine singola
del sistema



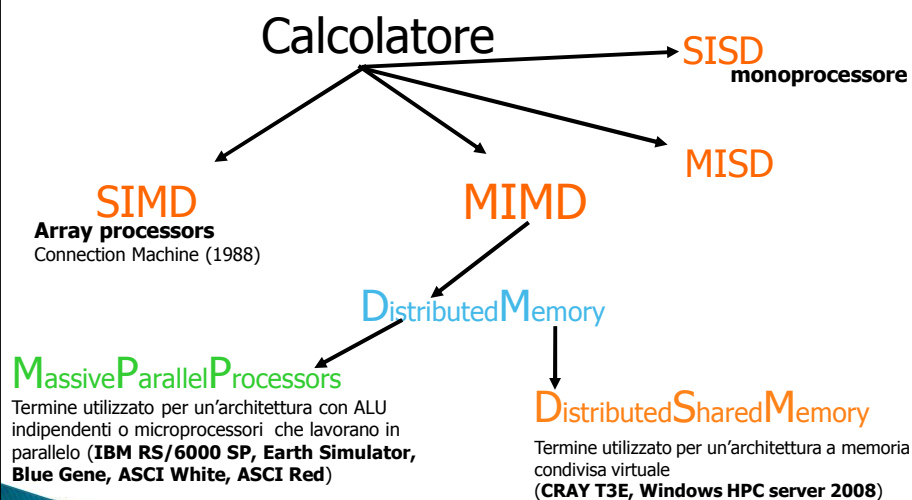
ASCI Red

Architettura MPP *Intel TeraFLOPS*
9632 processori
Topologia a griglia
Performance sostenuta: 2.3 TFlop
Picco di Performance: 3.2 TFlop

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

69

Tassonomia di Flynn (dal 1965)



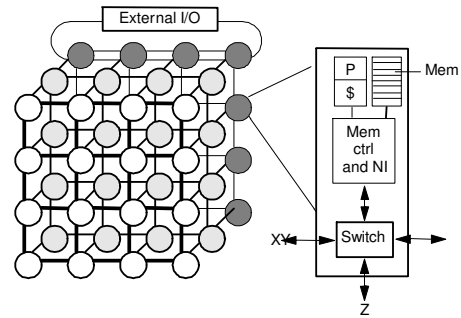
a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

70

Esempio MIMD-DSM



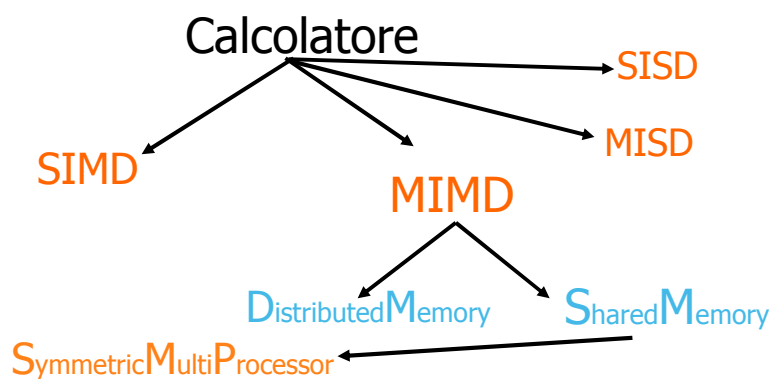
Cray T3E



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

71

Tassonomia di Flynn (dal 1965)

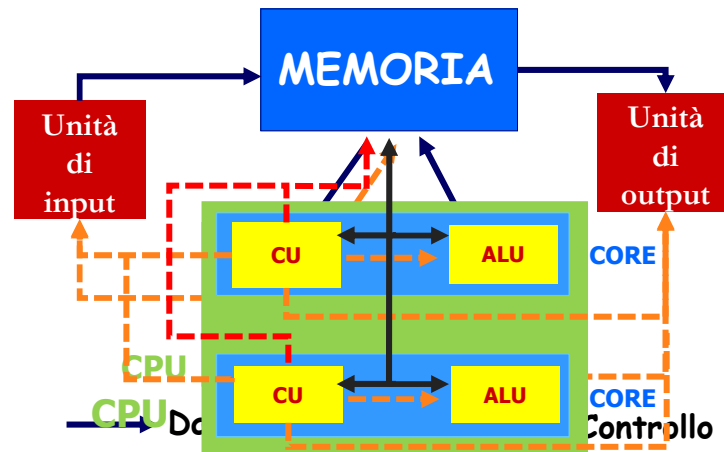


Termine utilizzato per un'architettura multiprocessore in cui due o più processori **identici** sono connessi ad una singola memoria principale condivisa.
(**MULTICORE INTEL**)

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

72

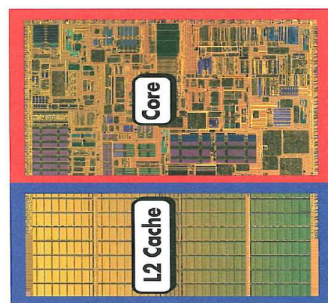
Esempio MIMD-SMP (multicore)



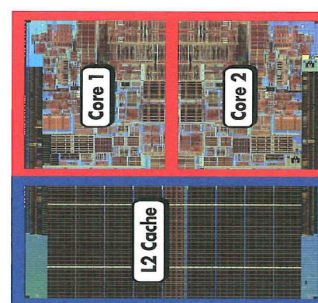
a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

73

Esempio MIMD-SMP (multicore)



Processore Intel Pentium M (single core)



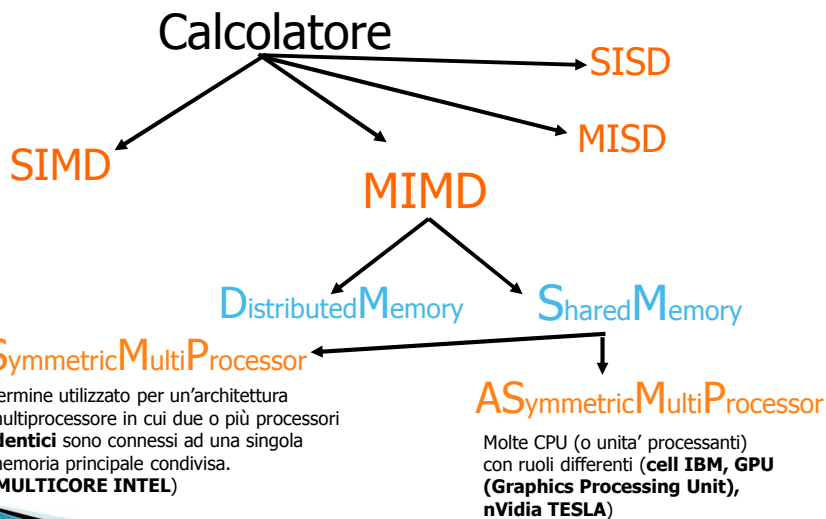
Processore Intel core duo

**Parallelismo
on-chip**

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

74

Tassonomia di Flynn (dal 1965)



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

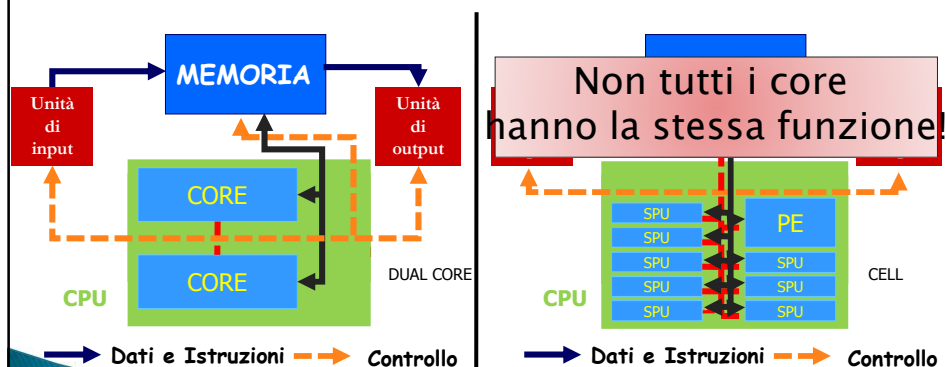
75

Esempio MIMD-ASMP

Cell IBM/Sony (PlayStation 3)

9 core

- 1 Processing Element (PE): architettura Risc con funzioni di controllo per gli altri 8
- 8 Synergistic Processing Unit (SPU): processore vettoriale per il calcolo

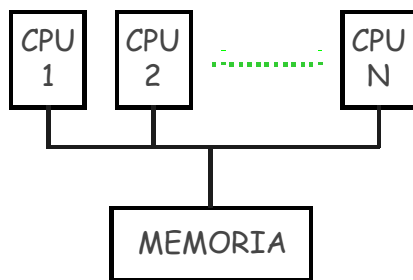


a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

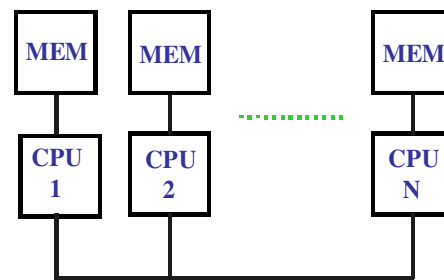
76

MIMD

Calcolatori MIMD a memoria **condivisa**
(shared-memory)



Calcolatori MIMD a memoria **distribuita**
(distributed-memory)

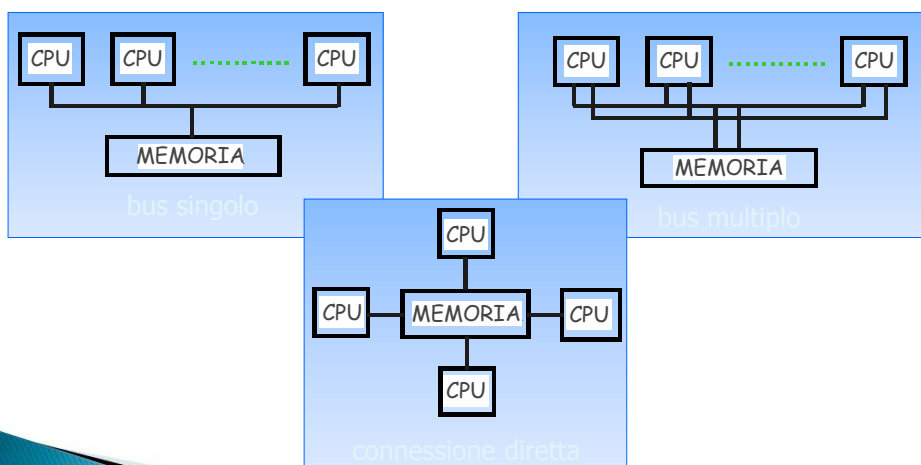


a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

77

MIMD: collegamento cpu-memoria

Calcolatori MIMD a **memoria condivisa**

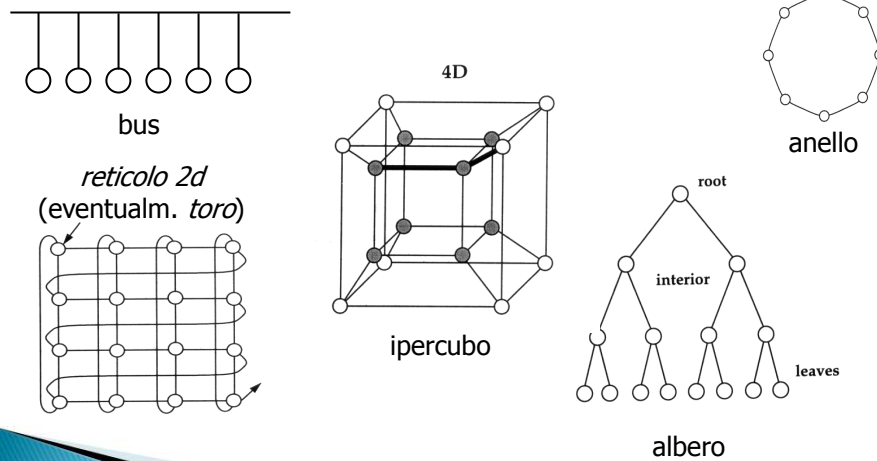


a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

78

MIMD: collegamento tra cpu

Calcolatori MIMD a **memoria distribuita**

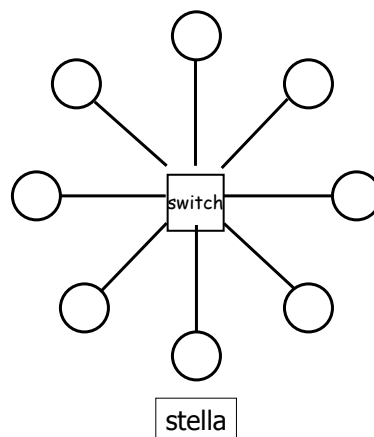


a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

79

MIMD: collegamento tra cpu

Calcolatori MIMD a **memoria distribuita**



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

80

Attualmente
tutti i processori
utilizzano forme diverse di
parallelismo, e nessun sistema in
commercio si può definire
puramente sequenziale.

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

81

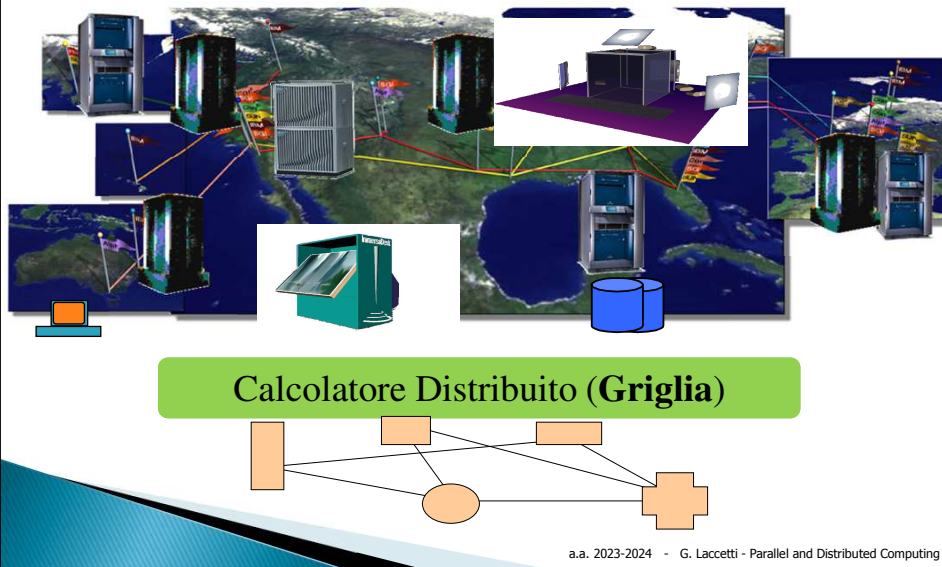
Introduzione al corso

»» Calcolo Distribuito

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

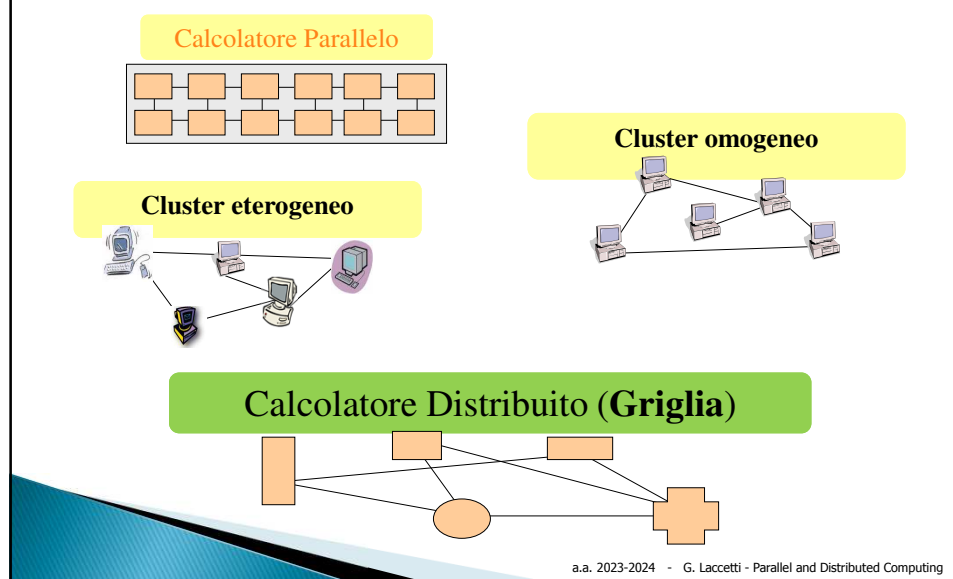
82

Calcolatore Distribuito



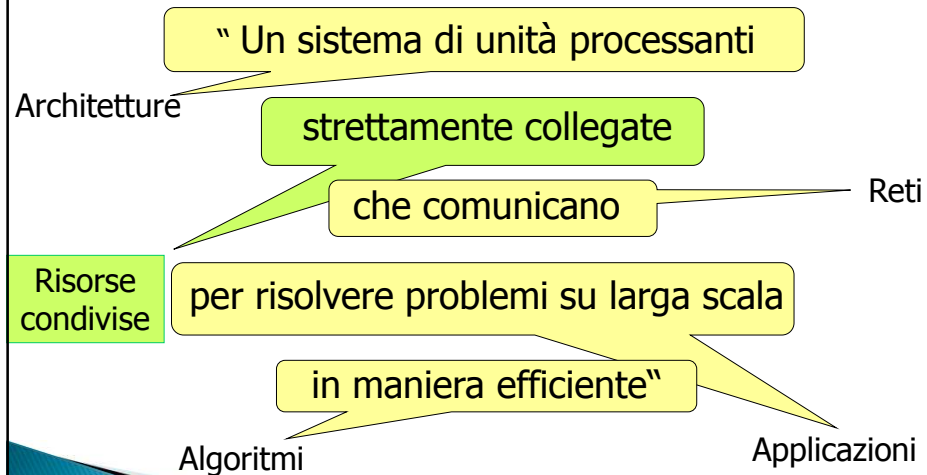
83

Calcolatore Distribuito



84

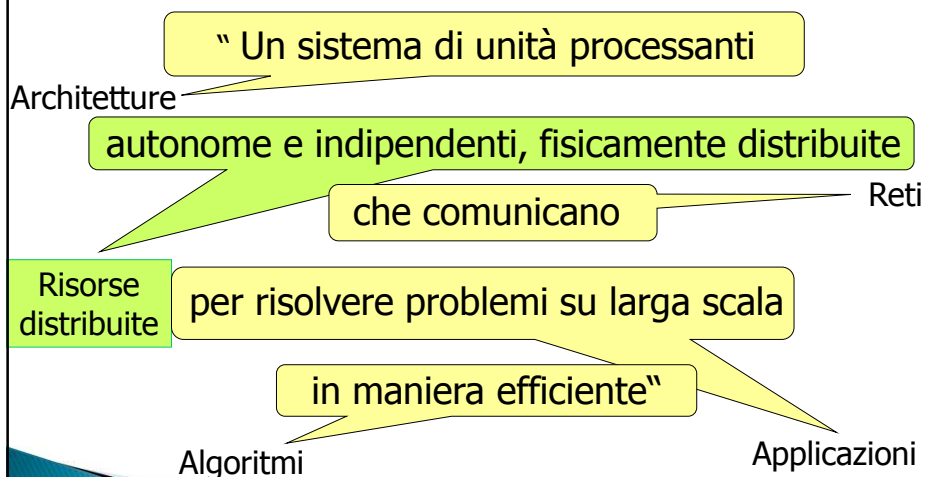
Ambiente di calcolo parallelo



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

85

Ambiente di calcolo distribuito



a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

86

Parallelo vs Distribuito

Calcolatore parallelo

↓
sistema di nodi collegati
da switch specializzati e
dedicati

↓
(tightly coupled systems)

Sistema ad arch.
distribuita

↓
sistema di nodi collegati da
reti geografiche

↓
(loosely coupled systems)

La differenza è nella rete di connessione

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

87

Parallelo vs Distribuito

Calcolatore parallelo

↓
Principale obiettivo:
Performance



Risorse di calcolo
omogenee

Sistema ad arch.
distribuita

↓
Principale obiettivo:
Ri-uso di risorse esistenti



Risorse di calcolo
eterogenee

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

88

Parallelo vs Distribuito

“in **parallel computing** we decompose into parts,
in **distributed computing** we assemble parts”

G.J. Fox, IEEE CiSE, 2002

“nel **calcolo parallelo** decomponiamo il problema,
nel **calcolo distribuito** assembliamo le risorse”

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

89

Calcolo parallelo per...

Ridurre il tempo necessario alla risoluzione computazionale
di un problema reale



“ wall - clock” time

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing

90

Calcolo distribuito per...

Riutilizzare “efficacemente” risorse hardware e software distribuite geograficamente sul territorio



“(ri)uso efficiente delle risorse”

a.a. 2023-2024 - G. Laccetti - Parallel and Distributed Computing