Calcolo Parallelo e Distribuito

matriceXvettore

1-2 strategie (approfondimenti)

Docente: Prof. L. Marcellino

Tutor: Prof. P. De Luca

PROBLEMA: Prodotto Matrice-Vettore

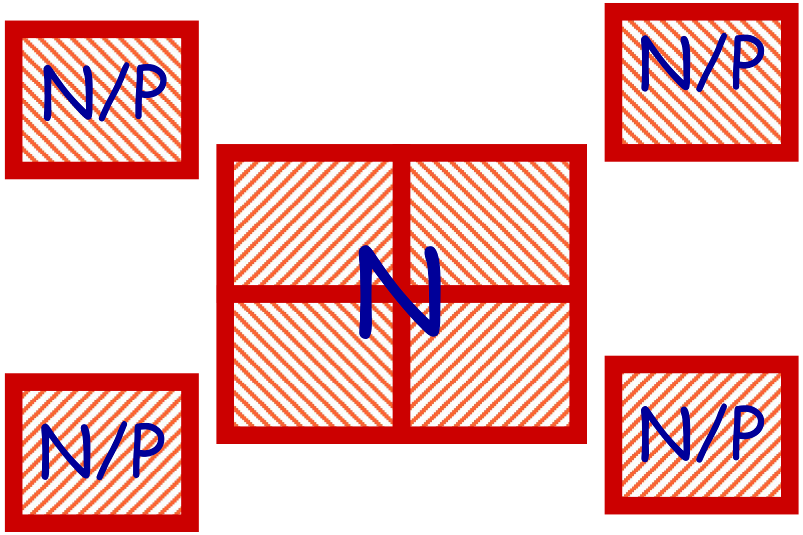
Progettazione
di un algoritmo parallelo
per architettura MIMD

per il calcolo del prodotto
di una matrice A per un vettore x:

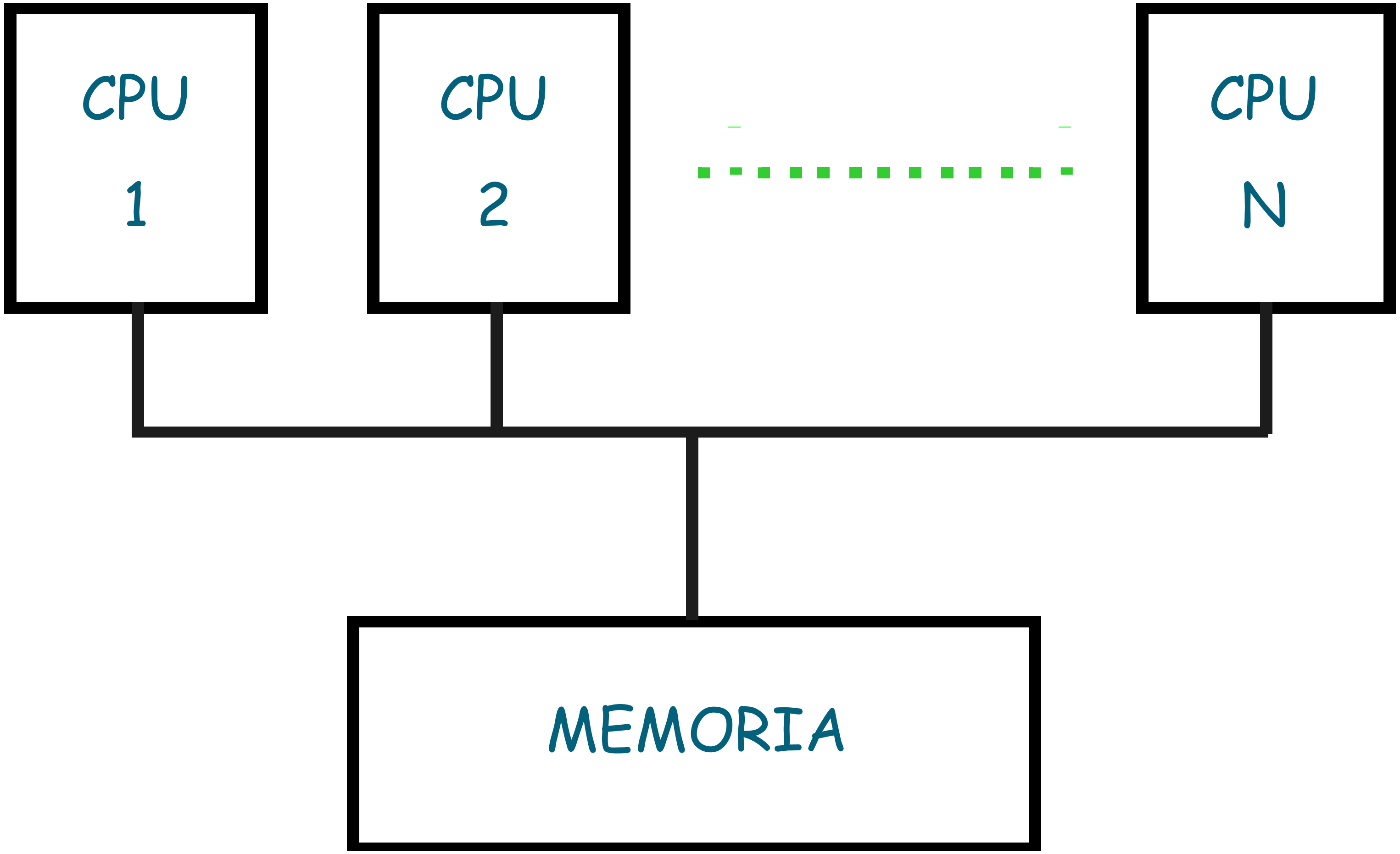
$$\boxed{Ax = y, \quad A \in R^{n \times n}, \quad x, y \in R^{n}}$$

DECOMPOSIZIONE: IDEA GENERALE

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



Schema Calcolatori
MIMD a memoria condivisa (shared-memory)



Approfondimenti:

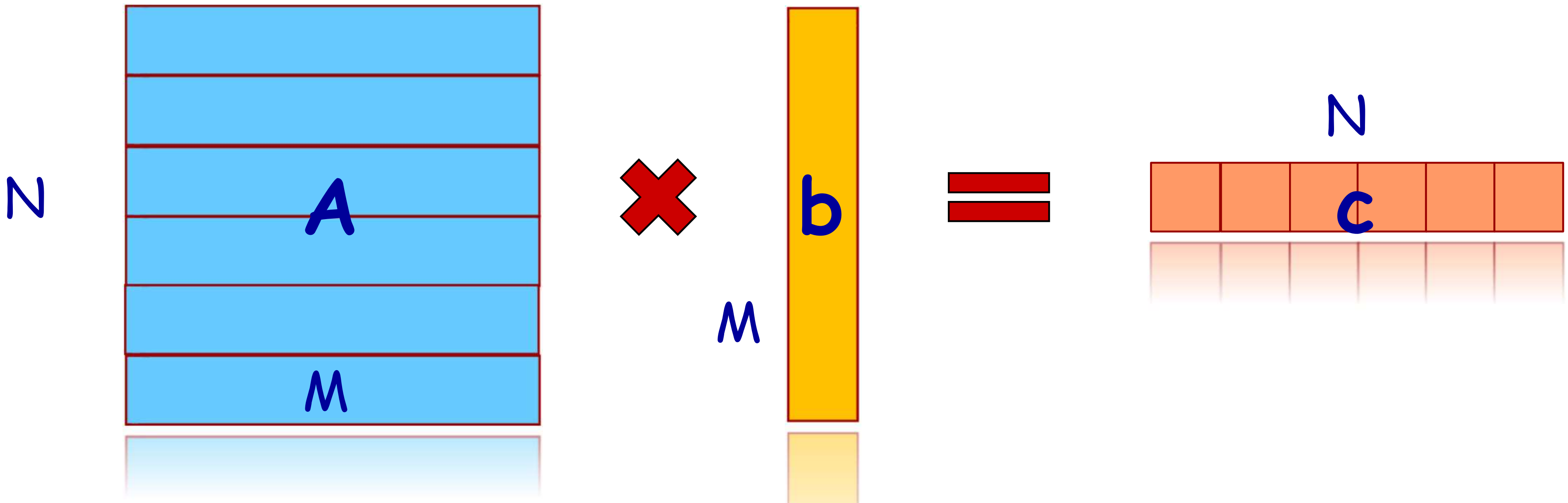
Prodotto Matrice-Vettore

algoritmo
per il calcolo del prodotto
di una matrice A per un vettore b:

matrice A: N righe, M colonne
Vettore b: M elementi

Complessità computazionale dell'algoritmo sequenziale

Matrice A: N righe, M colonne, Vettore b: M elementi



In sequenziale, **N prodotti scalari di lunghezza M**.

Per fare 1 prodotto scalare di lunghezza M, devo fare:

M molt + (M-1) add

Complessità computazionale dell'algoritmo sequenziale

Matrice A: N righe, M colonne - Vettore b: M elementi

In sequenziale, **N prodotti scalari di lunghezza M**, cioè:

N[M molt + (M-1) add]

molt ~ add

$$T_1(N \times M) = N[2M-1]$$

I STRATEGIA

Decomposizione 1

matrice A in

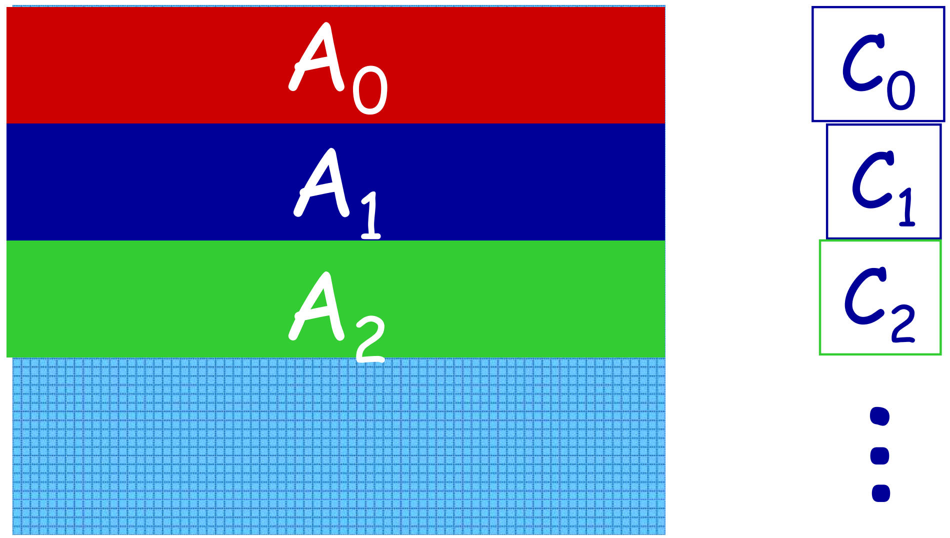
BLOCCHI di RIGHE

Calcolo **di speedup, overhead ed efficienza**
(def classica)

I Strategia: speed-up/efficienza (**def classica**)

Matrice A: N righe, M colonne - Vettore b: M elementi

$$T_1(N \times M) = N[2M-1]$$

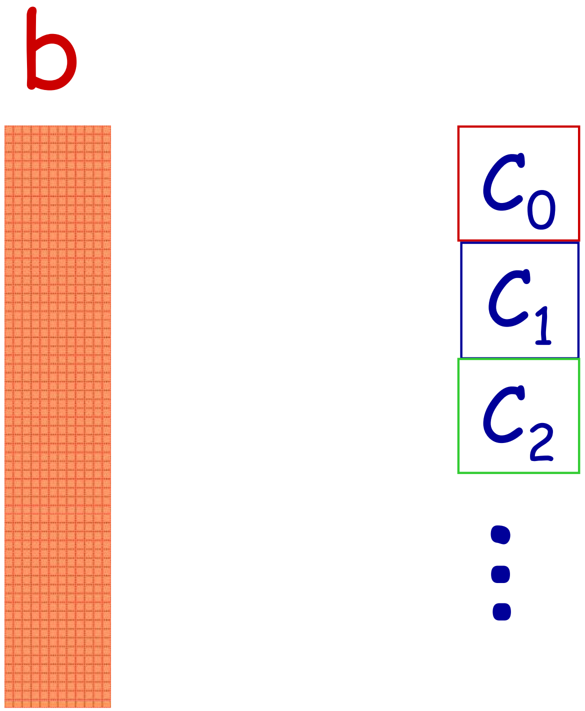


p core

I Strategia: speed-up/efficienza (**def classica**)

Matrice A: N righe, M colonne - Vettore b: M elementi

$$T_1(N \times M) = N[2M-1]$$



p core

I Strategia: speed-up/efficienza (**def classica**)

Matrice A: N righe, M colonne - Vettore b: M elementi

MIMD-SM

p core:

$$\dim[A_i] = (N/p) \times M; \quad \dim[b] = M$$

Tutti contemporaneamente, **N/p prodotti scalari di lunghezza M**, cioè:

$$T_p(N \times M) = N/p\ [\ 2M-1\ ]$$

I Strategia: speed-up/efficienza (**def classica**)

Matrice A: N righe, M colonne - Vettore b: M elementi

p core

$S_p(N\times M) = T_1(N\times M)/T_p(N\times M) =$

$= N[2M-1] /(N/p\ [\ 2M-1\ ]) = p\ \ N[2M-1] /(N[2M-1]) = p$

$Oh = p\ T_p(N\times M) - T_1(N\times M) =$

$= p(N/p\ [\ 2M-1\ ])\ - N[2M-1] = 0$

$E_p(N\times M) = S_p(N\times M)\ /p = p/p = 1$

I Strategia: **isoefficienza**

Matrice A: N righe, M colonne - Vettore b: M elementi

p core/processori

Oh = 0 -> $I(p_0 , p_1 , n_0)$ = 0/0 forma indeterminata

Per convenzione l'isoefficienza è posta uguale ad infinito, ovvero posso usare qualunque costante moltiplicativa per calcolare n_1 e quindi controllare la scalabilità dell'algoritmo.

Calcolo **di speedup ed efficienza**
(def Ware Amdahl-generalizzata)

I Strategia: speed-up/efficienza (**W-A**)

Matrice A: N righe, M colonne - Vettore b: M elementi

In sequenziale:

$T_1(N \times M)$ = N[2M-1] operazioni

Per calcolare lo speedup con la legge di W-A, la prima domanda che mi devo fare è **se per questa strategia di parallelizzazione posso esattamente distinguere la parte parallela**
(nella fase di calcolo locale lavorano tutti i processori)
~~e la parte sequenziale~~
~~(la collezione dei risultati avviene in maniera sequenziale)~~

SI

$$S_p = \frac{1}{\alpha + (1 - \alpha)/p}$$

I Strategia: speed-up/efficienza (**W-A**)

Matrice A: N righe, M colonne - Vettore b: M elementi

In sequenziale:

$T_1(N \times M)$ = N[2M-1] operazioni

In parallelo:

1 fase (tutta parallela)

Calcolo prodotti parziali

N/p [2M-1] operazioni

contemporaneamente

fatto da p processori/core

p N/p [2M-1]

delle N[2M-1] operazioni

$$1-\alpha = p\frac{N/p[2M-1]}{N[2M-1]} = 1 \Rightarrow \frac{1-\alpha}{p} = \frac{1}{p}$$

I Strategia: speed-up/efficienza (**W-A**)

Matrice A: N righe, M colonne - Vettore b: M elementi

In sequenziale:

$T_1(N \times M) = N[2M-1]$ operazioni

…e basta!

$$\boxed{\alpha = 0} \quad \boxed{\frac{1-\alpha}{p} = \frac{1}{p}}$$

$$\Rightarrow \quad \boxed{S(p) = \frac{1}{\frac{1}{p}} = p}$$

II STRATEGIA

Decomposizione 2

matrice A in

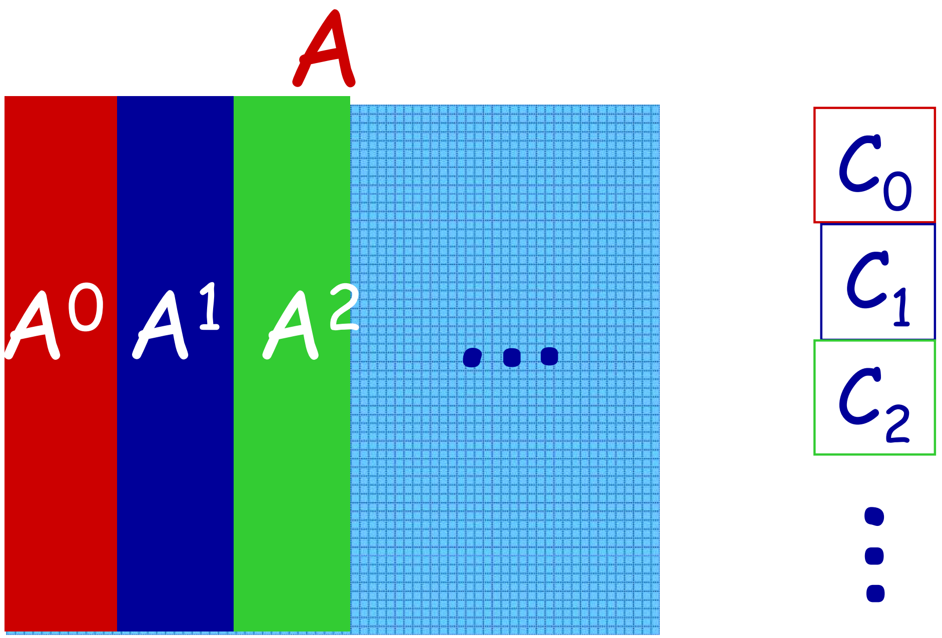
BLOCCHI di COLONNE

Calcolo **di speedup ed efficienza** **(def classica)**

II Strategia: speed-up/efficienza (**def classica**)

Matrice A: N righe, M colonne, Vettore b: M elementi

$$T_1(N \times M) = N[2M-1]$$

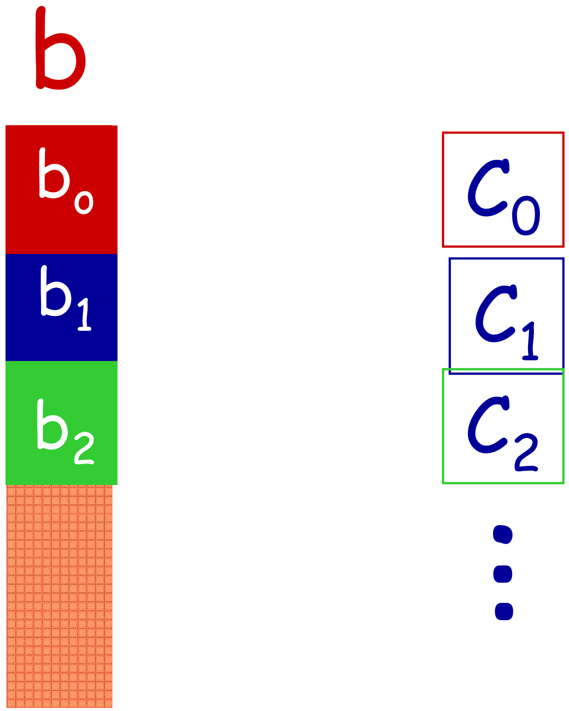


p core

II Strategia: speed-up/efficienza (**def classica**)

Matrice A: N righe, M colonne, Vettore b: M elementi

$$T_1(N \times M) = N[2M-1]$$



p core

II Strategia: speed-up/efficienza (**def classica**)

Matrice A: N righe, M colonne, Vettore b: M elementi

$\dim[A^i] = N \times (M/p)$

$\dim[b_i] = M/p$

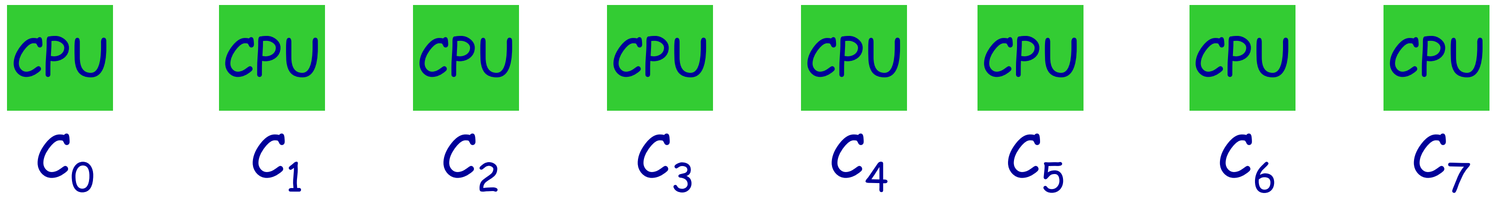
p core

Tutti contemporaneamente, **N prodotti scalari di lunghezza M/p**, cioè:

$$N\,[\,2M/p - 1\,]$$

Esempio p=8

p core



r_i vettori i=0,7, di lunghezza N

nell'unica memoria condivisa da sommare tra loro!!!

Come calcolare il risultato finale?

potremmo utilizzare una delle **due** strategie della somma per calcolare il risultato finale:

I strategia: ogni core somma (uno alla volta) i vettori calcolati dagli altri core
(componente per componente)

Esempio p=8

p core

CPU CPU CPU CPU CPU CPU CPU CPU

C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7

r_i vettori i=0,7, di lunghezza N

nell'unica memoria condivisa da sommare tra loro!!!

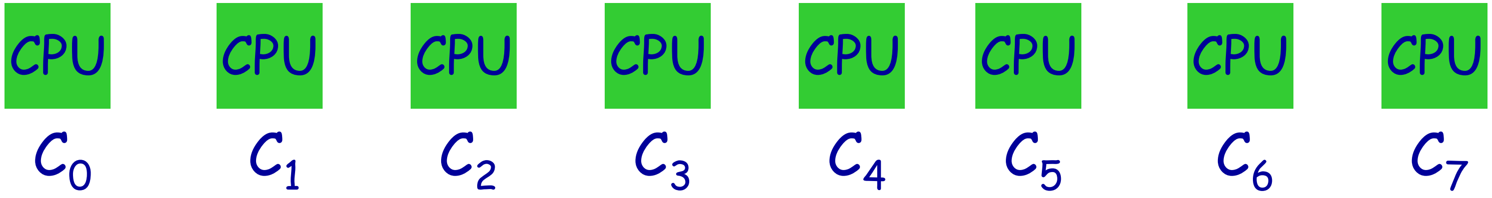
Come calcolare il risultato finale?

potremmo utilizzare una delle **due** strategie della somma per calcolare il risultato finale:

II strategia: [1 passo] contemporaneamente i core C_0 C_2 C_4 C_6 sommano il vettore di cui si sono occupati nella fase precedente con quelli calcolati dai core C_1 C_3 C_5 C_7 (componente per componente)

Esempio p=8

p core



r_i vettori i=0,7, di lunghezza N

nell'unica memoria condivisa da sommare tra loro!!!

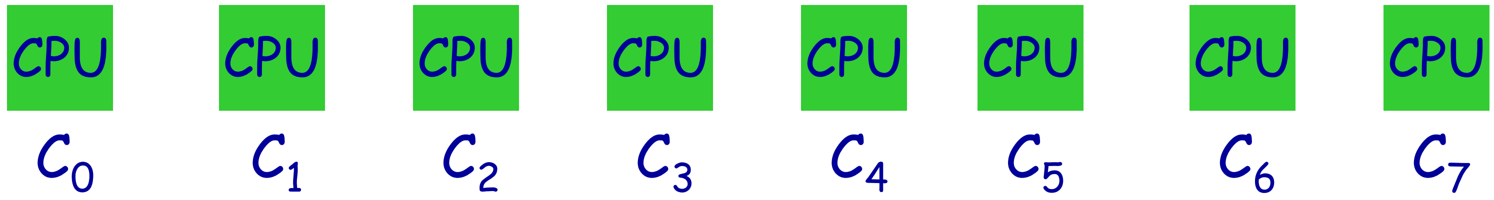
Come calcolare il risultato finale?

potremmo utilizzare una delle **due** strategie della somma per calcolare il risultato finale:

II strategia: [2 passo] contemporaneamente i core C_0 C_4 sommano il vettore di cui si sono occupati nella fase precedente con quelli calcolati dai core C_2 C_6 al **[1 passo]** (componente per componente)

Esempio p=8

p core



r_i vettori i=0,7, di lunghezza N

nell'unica memoria condivisa da sommare tra loro!!!

Come calcolare il risultato finale?

potremmo utilizzare una delle **due** strategie della somma per calcolare il risultato finale:

II strategia: [3 passo] il core C_0 somma il vettore di di cui si è occupato nella fase precedente con quello calcolato dal core C_4 al **[2 passo]** (componente per componente)

II Strategia: speed-up/efficienza (**def classica**)

Matrice A: N righe, M colonne, Vettore b: M elementi

p core

II strategia per collezione vettori

$S_p(N\times M) = T_1(N\times M)/T_p(N\times M) =$

$= N[2M-1] /(N [2M/p-1] + N \log_2(p))$

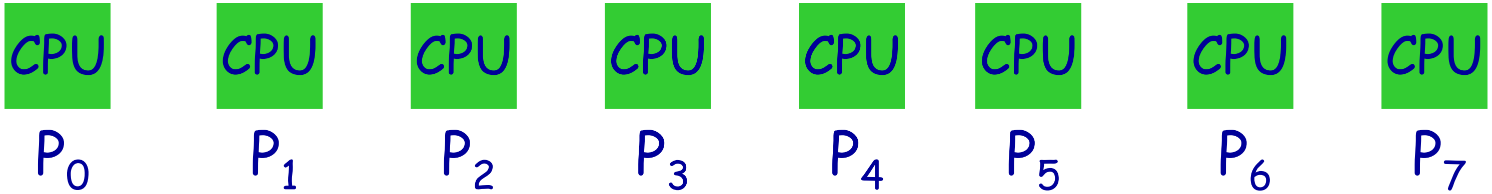
$Oh = p\, T_p(N\times M) - T_1(N\times M) =$

$= p\, (N [2M/p-1] + N \log_2(p)) - N[2M-1]$

$E_p(N\times M) = S_p(N\times M) /p =$

$= p\, N[2M-1] /(N [2M/p-1] + N \log_2(p))$

Esempio p=8



I strategia?

II Strategia: speed-up/efficienza (**def classica**)

Matrice A: N righe, M colonne, Vettore b: M elementi

p core

I strategia per collezione vettori

$S_p(N \times M) = T_1(N \times M)/T_p(N \times M) =$

$= N[2M-1] /(N [2M/p-1] + (p-1)N$

$Oh = p\, T_p(N \times M) - T_1(N \times M) =$

$= p\, (N [2M/p-1] + p-1\, (N)) - N[2M-1]$

$E_p(N \times M) = S_p(N \times M) /p =$

$= N[2M-1] /p\, (N [2M/p-1] + (p-1)N)$

II Strategia: **isoefficienza**

Matrice A: N righe, M colonne, Vettore b: M elementi

p core/processori

Per il calcolo dell'isoefficienza è necessario separare i conti tra la I e la II strategia impiegata per la collezione dei risultati!!!

I strategia per collezione vettori

$$Oh = p\,(N\,[\,2M/p-1\,] + (p-1)\,N) - N[2M-1] =$$

$$= pN[2M/p] - pN + p^2 N - pN - N[2M-1] =$$

$$= 2NM - pN + p^2 N - pN - 2NM + N =$$

$$= -2p\,N + p^2 N + N = N(-2p+p^2+1)$$

Overhead dipende dal numero delle righe e dal numero delle unità processanti!

II Strategia: **isoefficienza**

Matrice A: N righe, M colonne, Vettore b: M elementi

p core/processori

I strategia per collezione vettori

$$I(p_0, p_1, n_0) = C = [N_1(-2p_1 + p_1^2 + 1)]/[N_0(-2p_0 + p_0^2 + 1)]$$

$$\cancel{N_1} M_1 = C N_0 M_0 =$$

$$= \cancel{N_0} M_0 [\cancel{N_1}(-2p_1 + p_1^2 + 1)]/[\cancel{N_0}(-2p_0 + p_0^2 + 1)]$$

Nel calcolo delle nuove dimensioni N ed M, devo fissare il numero di righe e calcolare le colonne

II Strategia: **isoefficienza**

Matrice A: N righe, M colonne, Vettore b: M elementi

p core/processori

Rifare i conti per la seconda strategia...

II strategia per collezione vettori

$$Oh = p\,(N\,[\,2M/p-1\,] + N\log_2(p)) - N[2M-1] =$$

$$= 2M\,N - p\,N + p\,N\log_2(p) - 2M\,N + N =$$

$$= -p\,N + p\,N\log_2(p) + N =$$

$$= N\,(-p + p\log_2(p) + 1)$$

Anche in questo caso l'Overhead dipende dal numero delle righe e dal numero delle unità processanti!

II Strategia: **isoefficienza**

Matrice A: N righe, M colonne, Vettore b: M elementi

p core/processori

II strategia per collezione vettori

$$I(p_0,\ p_1,\ n_0) = C = [N_1(-p_1+p_1\ \log_2(p_1)+1)]/[N_0(-p_0+p_0\log_2(p_0)+1)]$$

$$\cancel{N_1}\ M_1 = C\ N_0\ M_0 =$$

$$= \cancel{N_0}\ M_0\ [\cancel{N_1}(-p_1+p_1\ \log_2(p_1)+1)]/[\cancel{N_0}(-p_0+p_0\log_2(p_0)+1)]$$

Stesse osservazioni di prima, nel calcolo delle nuove dimensioni, posso fissare un qualunque numero di righe e calcolare il più opportuno numero di colonne

Calcolo **di speedup ed efficienza**
(def Ware Amdahl-generalizzata)

II Strategia: speed-up/efficienza (W-A)

Matrice A: N righe, M colonne, Vettore b: M elementi

In sequenziale:

$T_1(N \times M) = N[2M-1]$ operazioni

Per calcolare lo speedup con la legge di W-A, la prima domanda che mi devo fare è **se per questa strategia di parallelizzazione posso esattamente distinguere la parte parallela**
(nella fase di calcolo locale lavorano tutti i processori)
e la parte sequenziale
(la collezione dei risultati avviene in maniera sequenziale)

I strategia per collezione vettori ⇒

$$S_p = \frac{1}{\alpha + \frac{1-\alpha}{p}}$$

II strategia per collezione vettori ⇒

$$S_p = \frac{1}{\alpha_1 + \sum_{k=2}^{p-1} \frac{\alpha_k}{k} + \frac{\alpha_p}{p}}$$

II Strategia: speed-up/efficienza (**W-A**)

Matrice A: N righe, M colonne, Vettore b: M elementi

In sequenziale:

$T_1(N \times M) = N[2M-1]$ operazioni

I strategia per collezione vettori

In parallelo:

1 fase (tutta parallela)

Calcolo prodotti parziali

N [2M/p -1] operazioni

contemporaneamente

fatto da p processori/core

p N [2M/p-1]

delle N[2M-1] operazioni

$$1-\alpha = p\frac{N\left[2M/p-1\right]}{N\left[2M-1\right]} \Rightarrow \frac{1-\alpha}{p} = \frac{p}{p}\cdot\frac{2M/p-1}{2M-1}$$

II Strategia: speed-up/efficienza (W-A)

Matrice A: N righe, M colonne, Vettore b: M elementi

In sequenziale:

$T_1(N \times M) = N[2M-1]$ operazioni

I strategia per collezione vettori

2 fase (tutta sequenziale): p-1 volte devo aggiornare il vettore in P_0 facendo delle somme componente per componente

$$\alpha = \frac{(p-1)N}{N[2M-1]} = \frac{p-1}{2M-1}$$

II Strategia: speed-up/efficienza (W-A)

Matrice A: N righe, M colonne, Vettore b: M elementi

I strategia aggiornamento

In sequenziale:

$T_1(N \times M) = N[2M-1]$ operazioni

$$\frac{1-\alpha}{p} = \frac{2M/p - 1}{2M-1}$$

$$\alpha = \frac{p-1}{2M-1}$$

$$S_p = \frac{1}{\dfrac{2M/p-1}{2M-1} + \dfrac{p-1}{2M-1}} = \frac{2M-1}{2M/p + p - 2}$$

II Strategia: speed-up/efficienza (**W-A**)

Matrice A: N righe, M colonne, Vettore b: M elementi

In sequenziale:

$T_1(N\times M)$ = N[2M-1] operazioni

II strategia
per collezione vettori

p=4

1 fase (tutta parallela)

$$\alpha_4 \equiv (1-\alpha) = p\,\frac{N\,(2M/p-1)}{N\,[2M-1]} = \frac{4\,(M/2-1)}{2M-1}$$

come prima, solo che ho sostituito p=4

II Strategia: speed-up/efficienza (W-A)

In sequenziale:

$T_1(N \times M) = N[2M-1]$ operazioni

II strategia
per collezione vettori

p=4

Nessuna fase in cui lavorano solo 3 processori/core

$$\boxed{\alpha_3 = 0}$$

Fase di parallelismo parziale (lavorano solo 2 processori/core):

$$\boxed{\alpha_2 = 2\frac{N}{N[2M-1]} = \frac{2}{2M-1}}$$

solo le somme componente per componente

II Strategia: speed-up/efficienza (W-A)

In sequenziale:

$T_1(N \times M) = N[2M-1]$ operazioni

II strategia
per collezione vettori

p=4

Fase puramente sequenziale

$$\alpha_1 = \frac{N}{N[2M-1]} = \frac{1}{2M-1}$$

L'ultima somma componente per componente

II Strategia: speed-up/efficienza (W-A)

In sequenziale:

$T_1(N \times M) = N[2M-1]$ operazioni

**II strategia
per collezione vettori**

p=4

Non resta che sostituire

$$\alpha_4 = \frac{4(M/2-1)}{2M-1} \qquad \alpha_3 = 0 \qquad \alpha_2 = \frac{2}{2M-1} \qquad \alpha_1 = \frac{1}{2M-1}$$

$$S_p = \frac{1}{\frac{1}{4}\cdot\frac{4(M/2-1)}{2M-1} + \frac{1}{2}\cdot\frac{2}{2M-1} + \frac{1}{2M-1}}$$

II Strategia: speed-up/efficienza (W-A)

In sequenziale:

$T_1(N \times M)$ = N[2M-1] operazioni

II strategia per collezione vettori

p=4

Non resta che sostituire

$$\boxed{\alpha_4 = \frac{4(M/2-1)}{2M-1}} \quad \boxed{\alpha_3 = 0} \quad \boxed{\alpha_2 = \frac{2}{2M-1}} \quad \boxed{\alpha_1 = \frac{1}{2M-1}}$$

$$\Rightarrow \boxed{S_p = \frac{1}{\frac{(M/2-1)}{2M-1} + \frac{1}{2M-1} + \frac{1}{2M-1}}}$$

II Strategia: speed-up/efficienza (**W-A**)

In sequenziale:

$T_1(N \times M) = N[2M-1]$ operazioni

II strategia
per collezione vettori

p=4

Non resta che sostituire

$$\alpha_4 = \frac{4(M/2-1)}{2M-1} \qquad \alpha_3 = 0 \qquad \alpha_2 = \frac{2}{2M-1} \qquad \alpha_1 = \frac{1}{2M-1}$$

$$S_p = \frac{1}{\frac{(M/2-1)}{2M-1} + \frac{2}{2M-1}} = \frac{2M-1}{M/2+1}$$