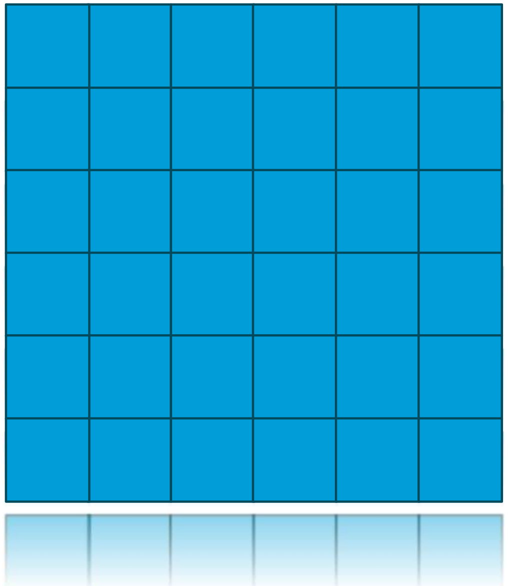
Calcolo Parallelo e Distribuito

Decomposizione dati 2D:
Algoritmi full-parallel per la gestione di matrici
valutazione strategie

Docente: Prof. L. Marcellino
Tutor: Prof. P. De Luca

Decomposizione di matrici

Prodotto di uno scalare con una matrice di grandi dimensioni!



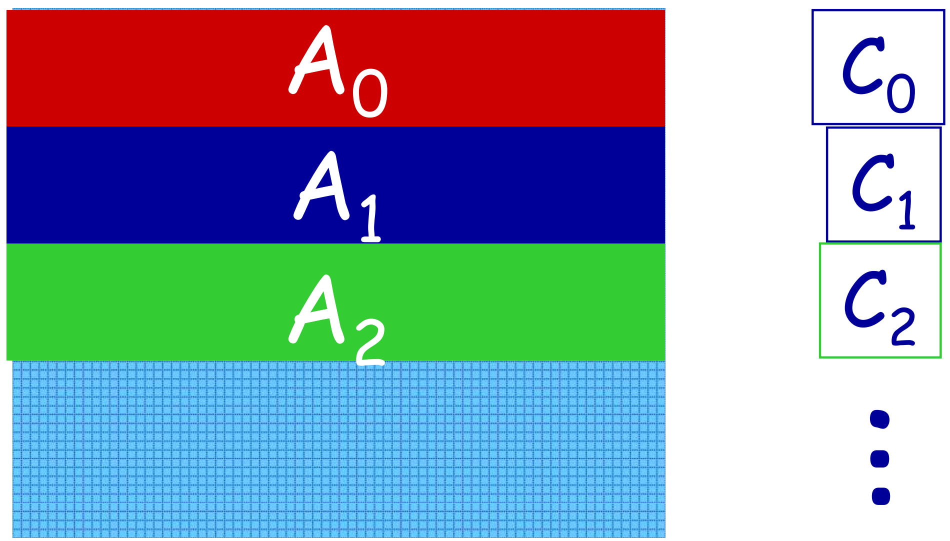
Input: $\beta \cdot A$: $dim(A)=N \times N$

Output: $C=\{c_{i,j}\} =\{\beta \cdot a_{i,j}\}$ $i=0,\dots,N-1$, $j=0,\dots,N-1$

I STRATEGIA

Suddividiamo la matrice A in

BLOCCHI di RIGHE



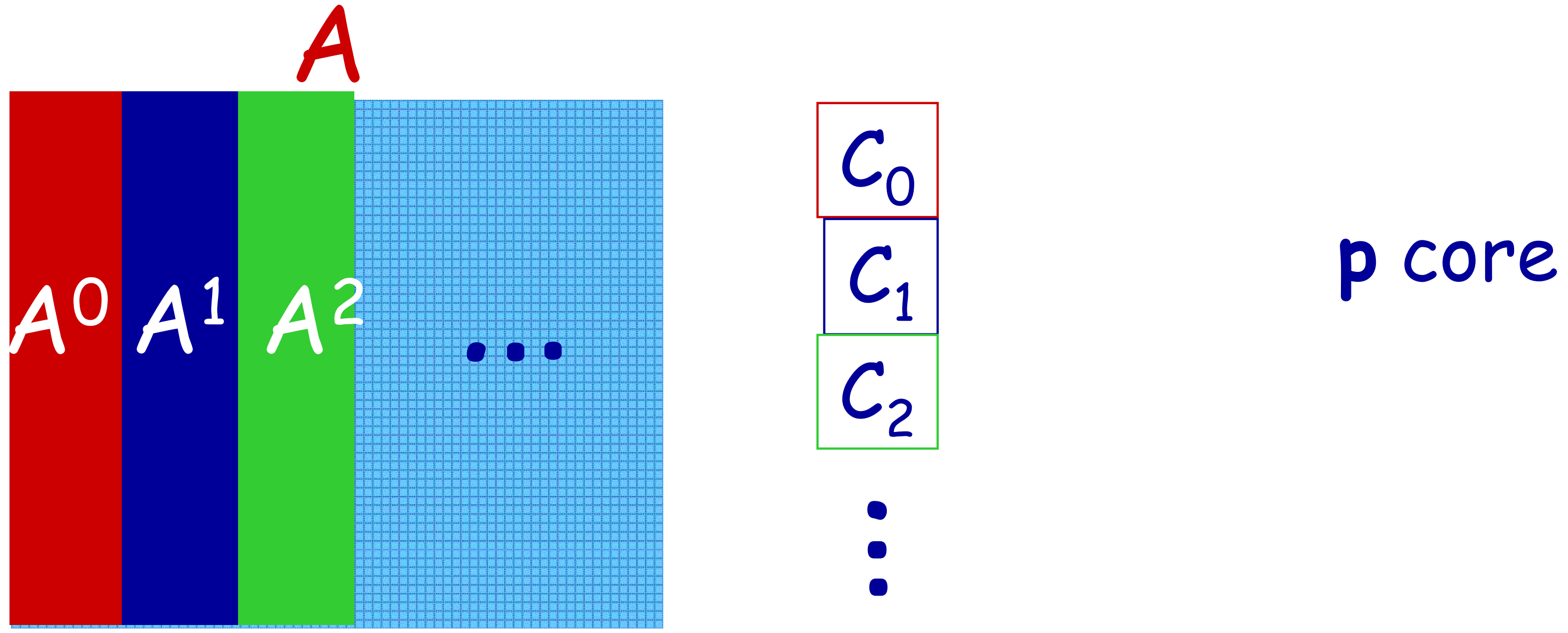
p core

II STRATEGIA

Suddividiamo la matrice

A in

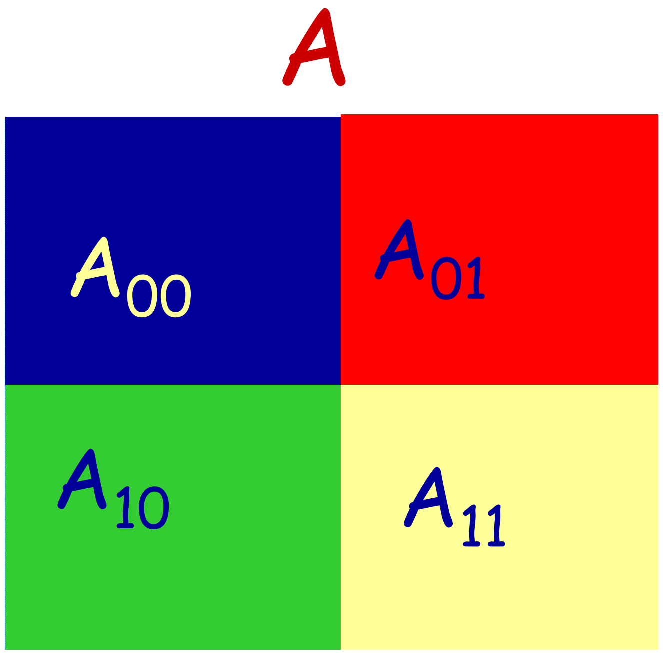
BLOCCHI di COLONNE



III STRATEGIA

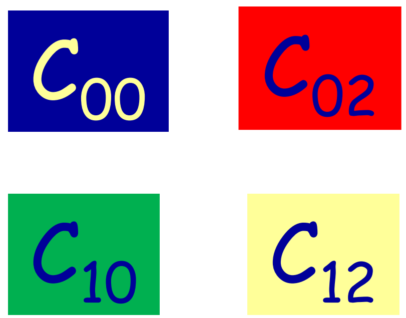
Suddividiamo la matrice *A* in

BLOCCHI di RigheColonne



GRIGLIA

qxp core



Strategie di parallelizzazione per problemi di tipo element-wise

Per ognuna delle strategie considerate (I, II, III) si tratta di un PROBLEMA COMPLETAMENTE PARALLELIZZABILE

FULL PARALLEL

nessuna collezione dei risultati

- Somma, prodotto puntuale, sottrazioni di matrici
- Somma, differenza, divisione con scalare
- Calcolo della trasposta, riflessione

con queste prime semplici operazioni è possibile fare filtri base a qualunque tipo di segnale

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

algoritmo parallelo per il prodotto di uno scalare per una matrice

prodotto di uno scalare per una matrice

$\dim[A]=N\times N$

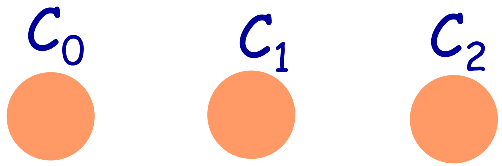
Complessità computazionale sequenziale

$N \times N$ prodotti

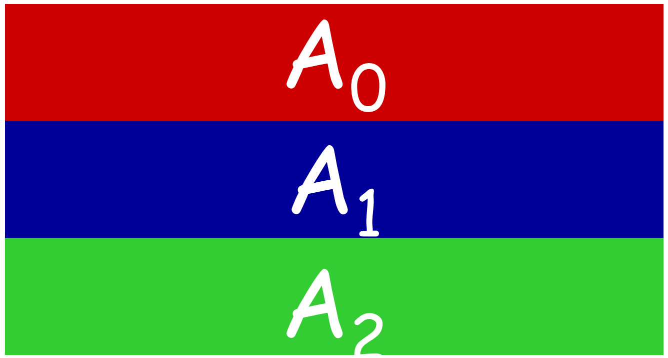
operazioni

$$T_1(N^2) = N^2$$

$p=3$, $\dim[A]=N\times N$



prodotto di uno scalare per una matrice



...

1 strategia – p righe – $\dim[A_{loc}]=(N/p)\times N$

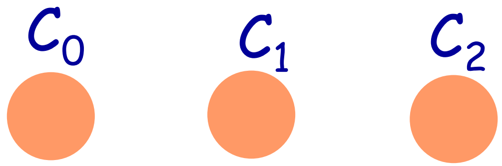
Calcolo prodotti locali

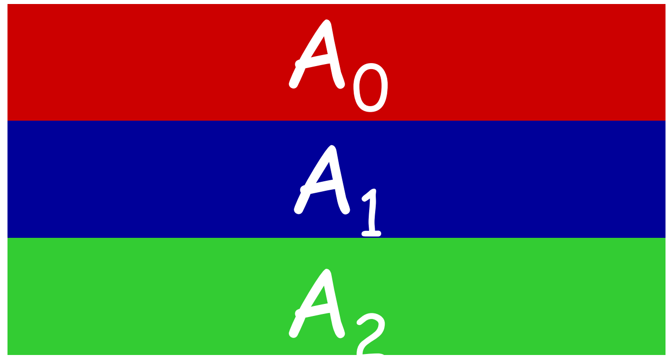
$(N/p)\times N$ prodotti

$$T_p(N^2) = N^2/p$$

prodotto di uno scalare per una matrice

$p=3$, $dim[A]=N\times N$





1 strategia – p righe – $dim[A_{loc}]=(N/p)\times N$

...

In sequenziale

$$T_1(N^2) = N^2$$

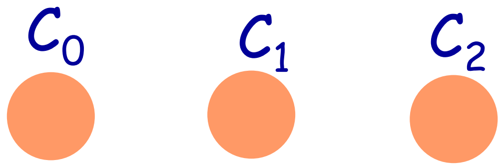
$$S_p = T_1(N^2)/T_p(N^2) = N^2/(N^2/p) = p$$

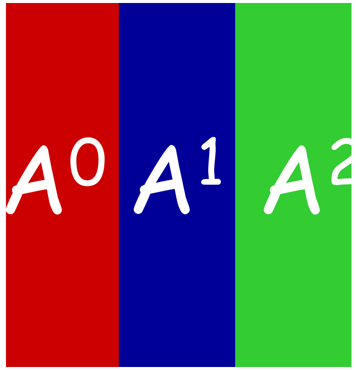
$$Oh = p\,T_p(N^2) - T_1(N^2) = p[N^2/p] - N^2 = 0$$

$$E_p = S_p/p = 1$$

prodotto di uno scalare per una matrice

$p=3$, $\dim[A]=N\times N$





2 strategia – p colonne – $\dim[A_{loc}] = N \times (N/p)$

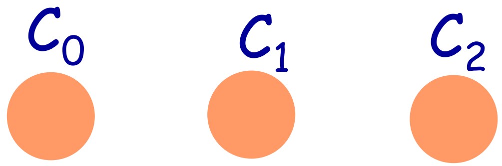
Calcolo prodotti locali

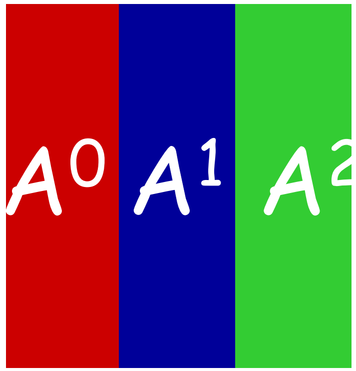
$N \times (N/p)$ **prodotti**

$$T_p(N^2) = N^2/p$$

prodotto di uno scalare per una matrice

$p=3$, $\dim[A]=N\times N$





2 strategia – p colonne – $\dim[A_{loc}] = N \times (N/p)$

In sequenziale

$$T_1(N^2) = N^2$$

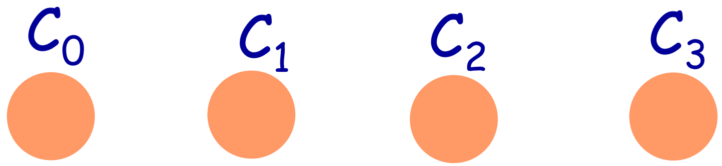
$$S_p = T_1(N^2)/T_p(N^2) = N^2/(N^2/p) = p$$

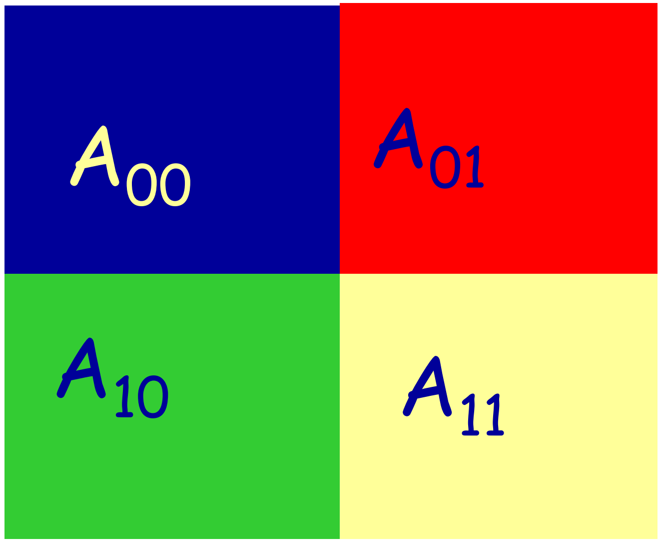
$$Oh = p\,T_p(N^2) - T_1(N^2) = p[N^2/p] - N^2 = 0$$

$$E_p = S_p/p = 1$$

$q \times p = 2 \times 2$, $\dim[A] = N \times N$

prodotto di uno scalare per una matrice





3 strategia – q righe - p colonne
$\dim[A_{loc}] = (N/q) \times (N/p)$

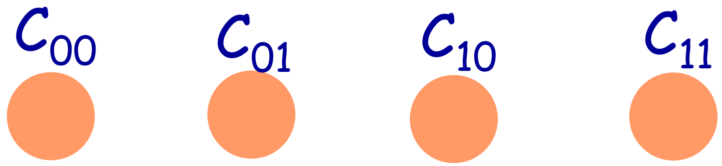
Calcolo prodotti locali

$(N/q) \times (N/p)$ prodotti

$$T_{qp}(N^2) = (N/q)\,(N/p)$$

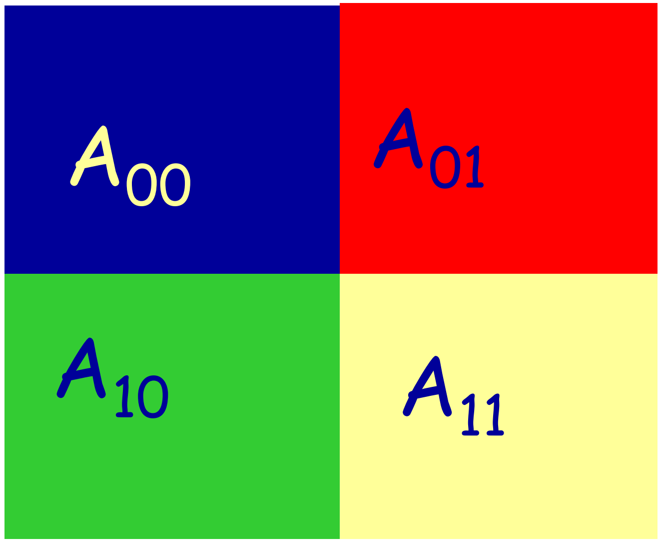
$q \times p = 2 \times 2$, $\dim[A] = N \times N$

prodotto di uno scalare per una matrice



3 strategia - q righe - p colonne

$\dim[A_{loc}] = (N/q) \times (N/p)$



In sequenziale

$$T_1(N^2) = N^2$$

$$S_{qp} = T_1(N^2)/T_{qp}(N^2) = N^2 / [(N/q)\ (N/p)] = qp$$

$$Oh = qp\ T_{qp}(N^2) - T_1(N^2) = qp[N^2/qp] - N^2 = 0$$

$$E_p = S_p / p = 1$$

Osservazioni:

Qual è la migliore strategia?

- **Finchè le matrici restano quadrate non c'è preferenza nella scelta tra 1 e 2 strategia!!!**

- **La terza strategia può essere oggetto di discussione:**

1. **quando q=p al denominatore dello speedup c'è un quadrato**
2. **Quando q≠p**

- **…**

isoefficienza

Definizione

L'ISOEFFICIENZA

è una funzione di tre variabili p_0 , p_1 , n_0

e definisce la costante che lega la nuova dimensione del problema da scegliere n_1 per valutare la **scalabilità** di un algortimo

$$I(p_0\ , p_1\ , n_0) = O_h(p_1\ , n_1\ )/O_h(p_0\ , n_0\ )$$

Calcolo isoefficienza

sprodotto di uno scalare per una matrice A di dimensione N^2

1 strategia – p righe 2 strategia – p colonne

$$O_h(p, N^2) = T_1(N^2) - p\ T_p(N^2) = N^2 - p\,(N^2/p) = 0$$

3 strategia – pq blocchi righe&colonne

$$O_h(pq, N^2) = T_1(N^2) - pq\ T_p(N^2) =$$

$$N^2 - pq\,(N^2/pq) = 0$$

$$\boxed{I = 0/0}$$

Forma indeterminata.

Per convenzione: I può assumere qualunque valore

Gli algoritmi full parallel sono *naturalmente scalabili*

Osservazioni:

Qual è la migliore strategia?

- Tutte le strategie sono ugualmente e naturalmente scalabili
- Finchè le matrici restano quadrate non c'è preferenza nella scelta tra 1 e 2 strategia!!!

- La terza strategia può essere oggetto di discussione:

1. quando q=p al denominatore dello speedup c'è un quadrato
2. Quando q≠p

- …

Caratterizziamo ora lo **speedup**
usando la legge di
Ware-Amdahl

parallelizzo tutto!

p=8, dim[A]=32×32

C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7

In sequenziale 32×32=1024 prodotti

1 strategia – p=8 righe

1 fase (tutta parallela)

Calcolo prodotti parziali
nloc = 4×32=128 prodotti

128 prodotti fatti contemporaneamente da 8 processori/core

128×8=1024 dei 1024 prodotti

$$(1-\alpha) = 32\times 32/32\times 32$$

p=8, dim[A]=32×32

C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7

In sequenziale 32×32=1024 prodotti

2 strategia – p=8 colonne

1 fase (tutta parallela)

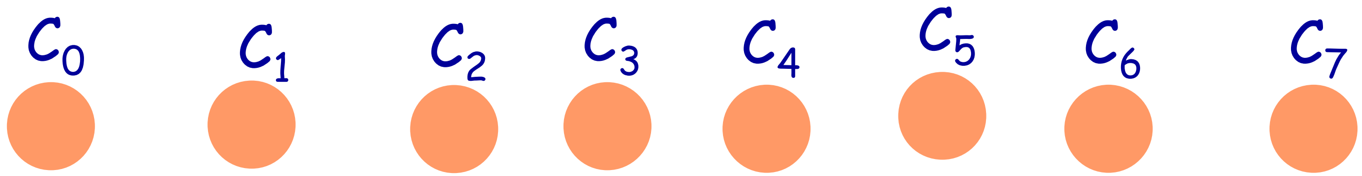
Calcolo prodotti parziali
nloc = 32×4=128 prodotti

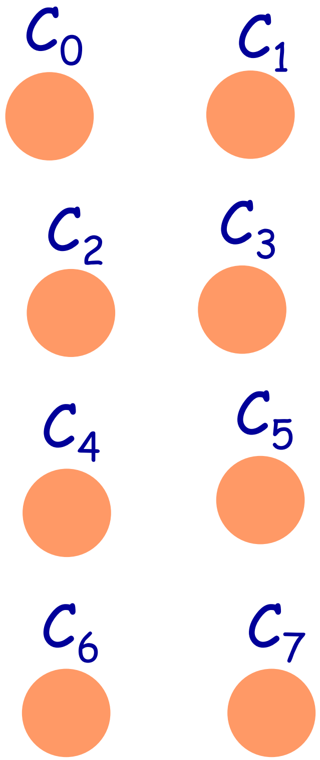
128 prodotti fatti contemporaneamente da 8 processori/core

128×8=1024
dei 1024 prodotti

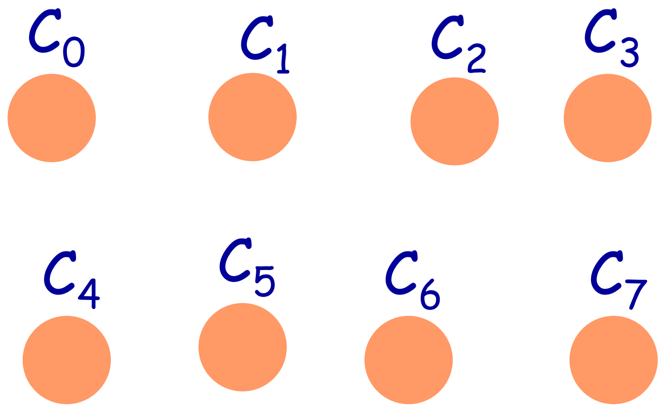
$$(1-\alpha) = 1024/1024$$

p=8, dim[A]=32×32



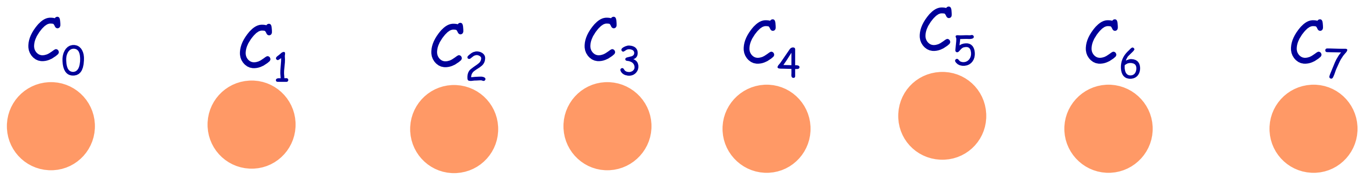


8=p×q=4×2



8=p×q=2×4

p=8, dim[A]=32×32



In sequenziale 32×32=1024 prodotti

3 strategia - p=4 righe q=2 colonne

1 fase (tutta parallela)

Calcolo prodotti parziali
nloc = 32/4×32/2=
=128 prodotti

128 prodotti fatti contemporaneamente da 8 processori/core

128×8=1024
dei 1024 prodotti

$$(1-\alpha) = 1024/1024$$

p=8, dim[A]=32×32

C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7

In sequenziale 32×32=1024 prodotti

3 strategia – p=2 righe q=4 colonne

1 fase (tutta parallela)

Calcolo prodotti parziali
nloc = 32/2×32/4=
=128 prodotti

128 prodotti fatti contemporaneamente da 8 processori/core

128×8=1024
dei 1024 prodotti

$$(1-\alpha) = 1024/1024$$

Osservazioni:

Qual è la migliore strategia?

- Tutte le strategie sono ugualmente e naturalmente scalabili
- Finchè le matrici restano quadrate non c'è preferenza nella scelta tra 1 e 2 strategia!!!
- Abbiamo dimostrato che quando $q \neq p$ non c'è nessuna differenza fra le 1-2-3 strategie.

- La terza strategia può essere oggetto di discussione:

1. quando $q=p$ al denominatore dello speedup c'è un quadrato

- … VEDIAMO QUELLO CHE SUCCEDE CON $p=2$, $q=2$

p=4, dim[A]=32×32

C_0 C_1 C_2 C_3

In sequenziale 32×32=1024 prodotti

1 strategia - p=4

1 fase (tutta parallela)

Calcolo prodotti parziali
nloc = 32/4×32=8×32
=256 prodotti

256 prodotti fatti contemporaneamente da 4 processori/core

256×4=1024
dei 1024 prodotti

$$(1-\alpha) = 1024/1024$$

$p=4$, $\dim[A]=32\times32$

C_0 C_1 C_2 C_3

In sequenziale $32\times32=1024$ prodotti

2 strategia – $p=4$

1 fase (tutta parallela)

Calcolo prodotti parziali

nloc = $32\times32/4=32\times8$

$=256$ prodotti

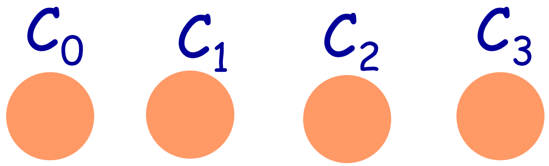
256 prodotti fatti contemporaneamente da 4 processori/core

$256\times4=1024$

dei 1024 prodotti

$$(1-\alpha)=1024/1024$$

p=4, dim[A]=32×32



In sequenziale 32×32=1024 prodotti

3 strategia – p=2, q=2

1 fase (tutta parallela)

Calcolo prodotti parziali
nloc = 32/2×32/2=16×16
=256 prodotti

256 prodotti fatti contemporaneamente da 4 processori/core

256×4=1024
dei 1024 prodotti

$$(1-\alpha) = 1024/1024$$

Osservazioni:

Qual è la migliore strategia?

- Tutte le strategie sono ugualmente e naturalmente scalabili
- Finchè le matrici restano quadrate non c'è preferenza nella scelta tra 1 e 2 strategia!!!
- Abbiamo dimostrato che quando q≠p non c'è nessuna differenza fra le 1-2-3 strategie.

- La terza strategia può essere oggetto di discussione:

1. quando q=p al denominatore dello speedup c'è un quadrato
2. … VEDIAMO QUELLO CHE SUCCEDE CON p=2, q=2

- non c'è nessuna differenza fra le 1-2-3 strategie.

Cosa succede se le matrici sono del tipo N ×M
e/o le dimensioni non sono
esattamente divisibili per p o q ???