Calcolo Parallelo e Distribuito

Prodotto Matrice-Vettore approfondimenti
Strategia 3 – parte1

Docente: Prof. L. Marcellino

Tutor: Prof. P. De Luca

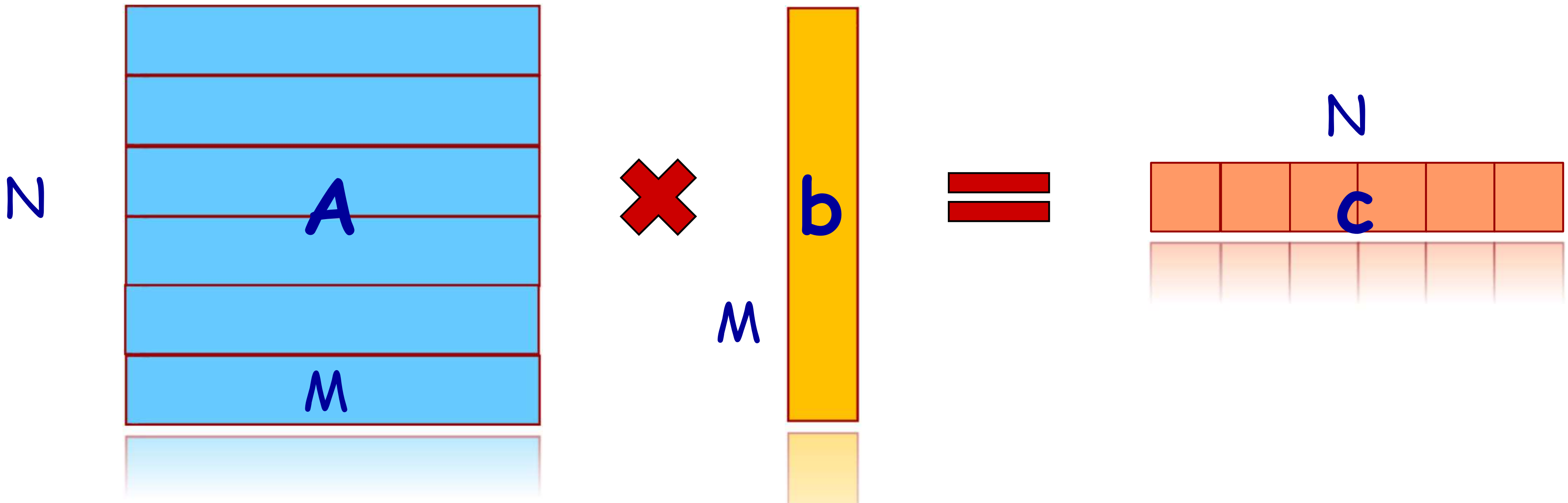
PROBLEMA: Prodotto Matrice-Vettore

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

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

PROBLEMA: Prodotto Matrice-Vettore

Progettazione
di un algoritmo parallelo
per architettura MIMD

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

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

III STRATEGIA

Decomposizione 1: BLOCCHI di RIGHE

+

Decomposizione 2: BLOCCHI di COLONNE

=

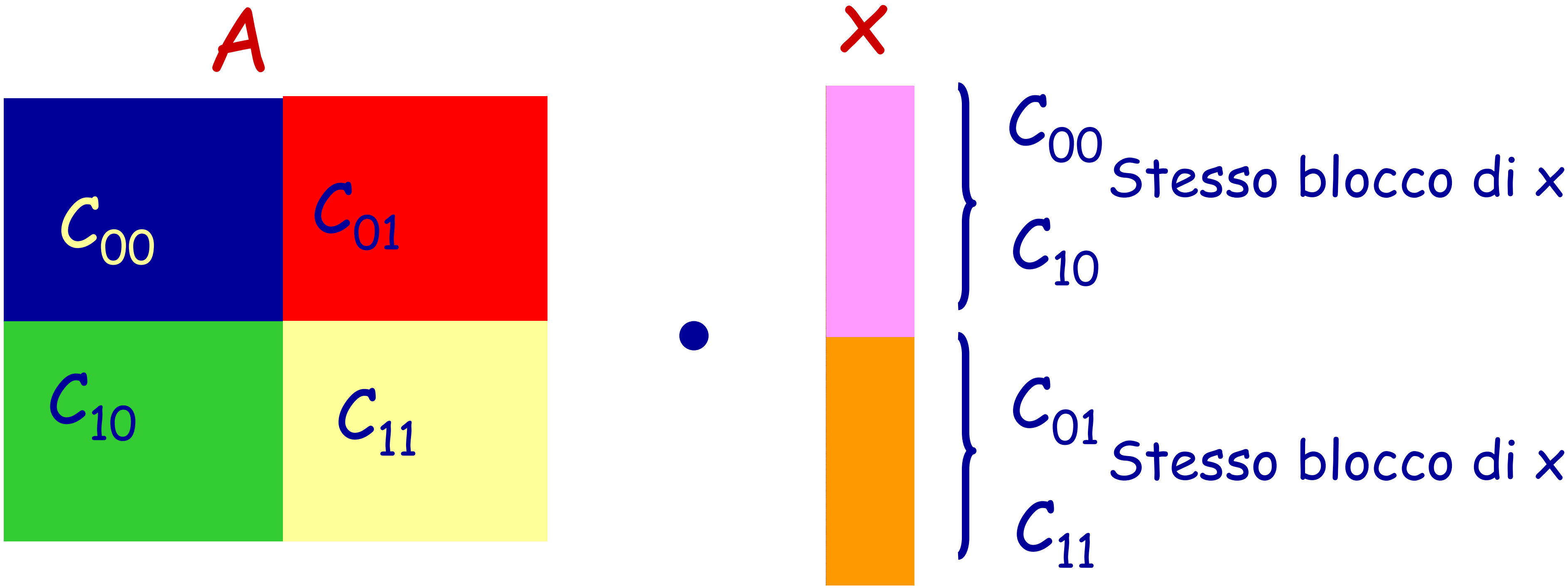
Decomposizione 3:

BLOCCHI RigheColonne

III Strategia: Esempio (4 core)

Distribuzione del lavoro sulla matrice *A*

per blocchi righe&colonne



Ogni core lavora col pezzo di x che gli serve

Esempio N = 6 , Core=4

C_{00}

$$a_{00} \cdot x_0 + a_{01} \cdot x_1 + a_{02} \cdot x_2$$
$$a_{10} \cdot x_0 + a_{11} \cdot x_1 + a_{12} \cdot x_2$$
$$a_{20} \cdot x_0 + a_{21} \cdot x_1 + a_{22} \cdot x_2$$

C_{01}

$$a_{03} \cdot x_3 + a_{04} \cdot x_4 + a_{05} \cdot x_5$$
$$a_{13} \cdot x_3 + a_{14} \cdot x_4 + a_{15} \cdot x_5$$
$$a_{23} \cdot x_3 + a_{24} \cdot x_4 + a_{25} \cdot x_5$$

C_{10}

$$a_{30} \cdot x_0 + a_{31} \cdot x_1 + a_{32} \cdot x_2$$
$$a_{40} \cdot x_0 + a_{41} \cdot x_1 + a_{42} \cdot x_2$$
$$a_{50} \cdot x_0 + a_{51} \cdot x_1 + a_{52} \cdot x_2$$

C_{11}

$$a_{33} \cdot x_3 + a_{34} \cdot x_4 + a_{35} \cdot x_5$$
$$a_{43} \cdot x_3 + a_{44} \cdot x_4 + a_{45} \cdot x_5$$
$$a_{53} \cdot x_3 + a_{54} \cdot x_4 + a_{55} \cdot x_5$$

Calcolo dei prodotti parziali

Esempio N = 6 , Core=4

C_{00}		C_{01}
r_0		s_0
r_1	+	s_1
r_2		s_2

C_{10}		C_{11}
r_3		s_3
r_4	+	s_4
r_5		s_5

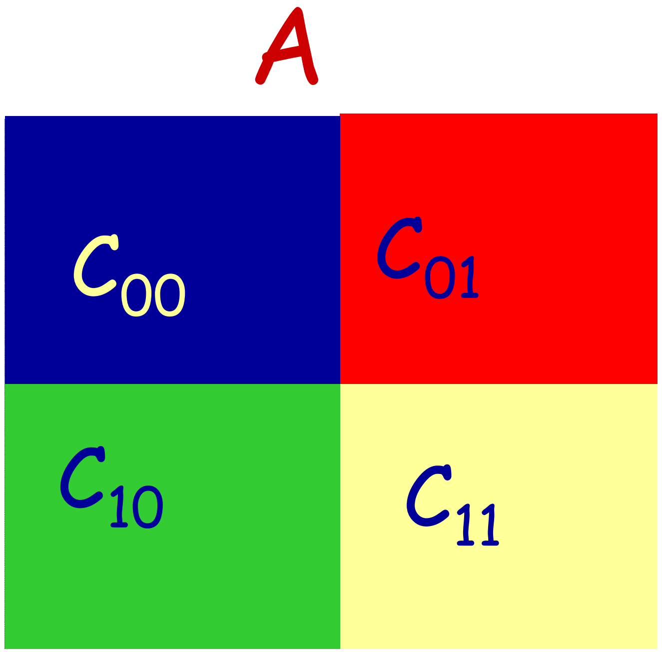
Coo e C10 sommano in parallelo

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

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

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

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



GRIGLIA

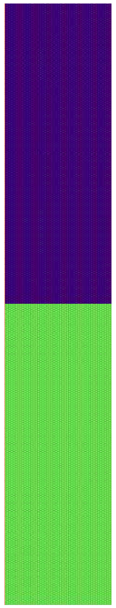
qxp=4=2x2 core

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

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

$$T_1(N \times M) = N[2M-1]\, t_{calc}$$

b



C_{00} Stesso blocco di b

C_{10}

C_{01} Stesso blocco di b

C_{11}

GRIGLIA

qxp=4=2x2 processori

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

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

$$dim[A_{loc}] = (N/q)x(M/p)$$

$$dim[b_{loc}] = M/p$$

GRIGLIA

qxp=4=2x2 core

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

N/q [2M/p-1] tcalc

N.B.: non è detto che la **GRIGLIA** sia **quadrata**, potrebbe essere anche **rettangolare qxp=8=2x4**

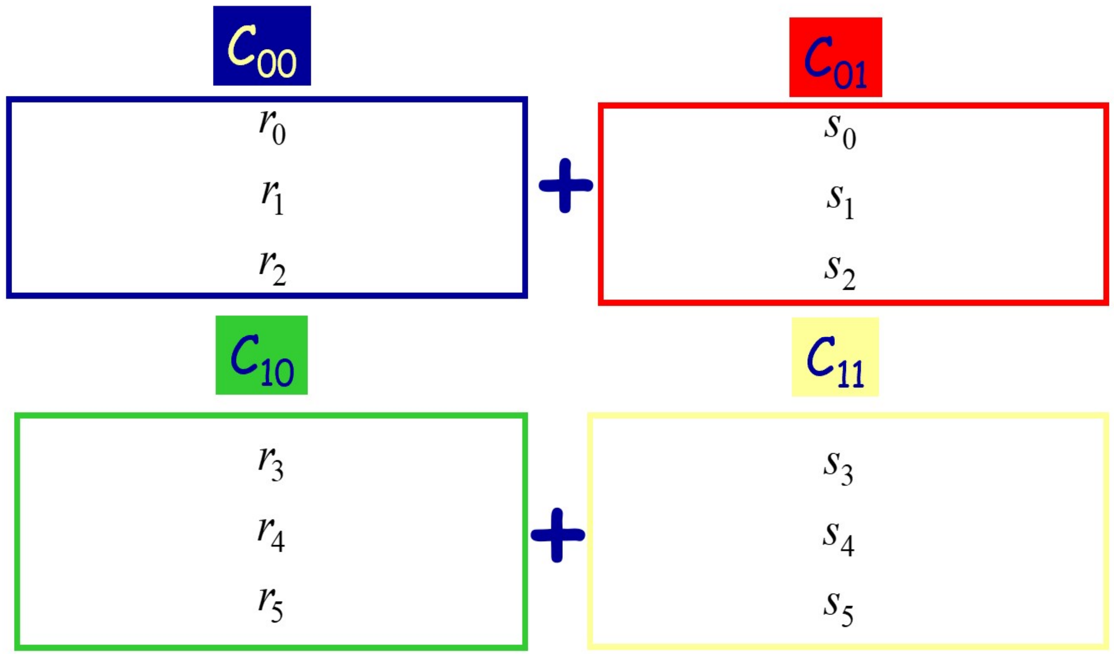
III Strategia: speed-up/efficienza (**def classica**)

Collezione dei risultati:

comunicazione e somma in parallelo

GRIGLIA

qxp=4=2x2 core



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

Collezione dei risultati

La scelta della strategia dipende da quanto vale p.

Se p è potenza di 2 allora conviene la II strategia, altrimenti (ed in particolare se è dispari) conviene usare la I strategia!!!

C_{00} + C_{01}

C_{10} + C_{11}

4=qxp=2x2 p=2

I strategia

I passo (e unico)

C_{00} C_{10} aggiornano il proprio vettore effettuando N/q somme, cioè:

$$N/q = N/2$$

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

Collezione dei risultati

$4=q\times p=2\times 2 \quad p=2$

I strategia

Ho finito.

La complessità computazionale della fase di collezione dei risultati locali vale…

… in generale:

$(p-1)\ N/q$

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

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

GRIGLIA qxp

I strategia per somma vettori

$S_{qxp}(NxM) = T_1(NxM)/T_{qxp}(NxM) =$

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

$Oh = qxp\ T_{qxp}(NxM) - T_1(NxM) =$

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

$E_{qxp}(NxM) = S_{qxp}(NxM) / (qxp) =$

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

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

Collezione dei risultati

La scelta della strategia dipende da quanto vale p.

Se p è potenza di 2 allora conviene la II strategia, altrimenti (ed in particolare se è dispari) conviene usare la I strategia!!!

C_{00} + C_{01} C_{02} + C_{03}

C_{10} + C_{11} C_{12} + C_{13}

8=qxp=2x4 p=4

II strategia

I passo

C_{00} C_{10} C_{02} C_{12} aggiornano con il proprio vettore effettuando N/q somme, cioè:

$$N/q = N/2$$

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

Collezione dei risultati

La scelta della strategia dipende da quanto vale p.

Se p è potenza di 2 allora conviene la II-III strategia, altrimenti (ed in particolare se è dispari) conviene usare la I strategia!!!

8=qxp=2x4 **p=4**

II strategia

C_{00}	+	C_{02}
C_{10}	+	C_{12}

II passo

C_{00} C_{10} aggiornano con il proprio vettore effettuando altre N/q somme, cioè:

$$N/q = N/2$$

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

Collezione dei risultati

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

$4=q \times p=2 \times 2$ $\quad$ $p=4$

II strategia

Ho finito.

La complessità computazionale della fase di collezione dei risultati locali vale…

… in generale:

$N/q \cdot \log_2(p)$

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

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

GRIGLIA qxp

II strategia per collezione vettori

$$S_{qxp}(NxM) = T_1(NxM)/T_{qxp}(NxM) =$$

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

$$Oh = qxp\; T_{qxp}(NxM) - T_1(NxM) =$$

$$= pxq\; (N/q\,[\,2M/p-1\,] + N/q\,\log_2(p)) - N[2M-1]\; t_{calc}$$

$$E_{qxp}(NxM) = S_{qxp}(NxM) \,/\, (qxp) =$$

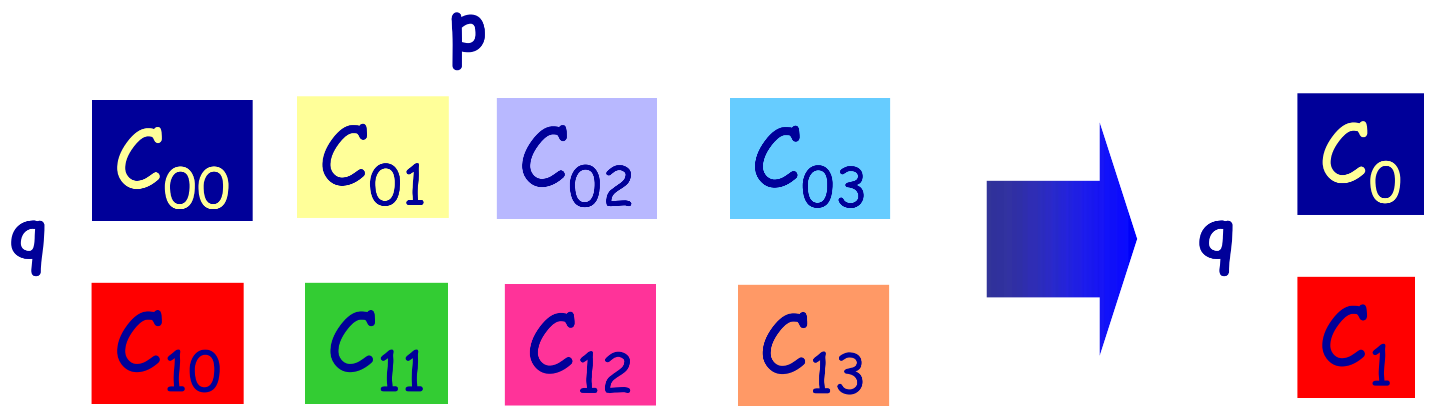
$$= N[2M-1] \,/\, [(qxp)(N/q\,[\,2M/p-1\,] + N/q\,\log_2(p))]$$

Speed-up/efficienza (**def classica**)

GRIGLIA qxp

Attenzione:

- è possibile riconoscere la I strategia dalla III quando p=1

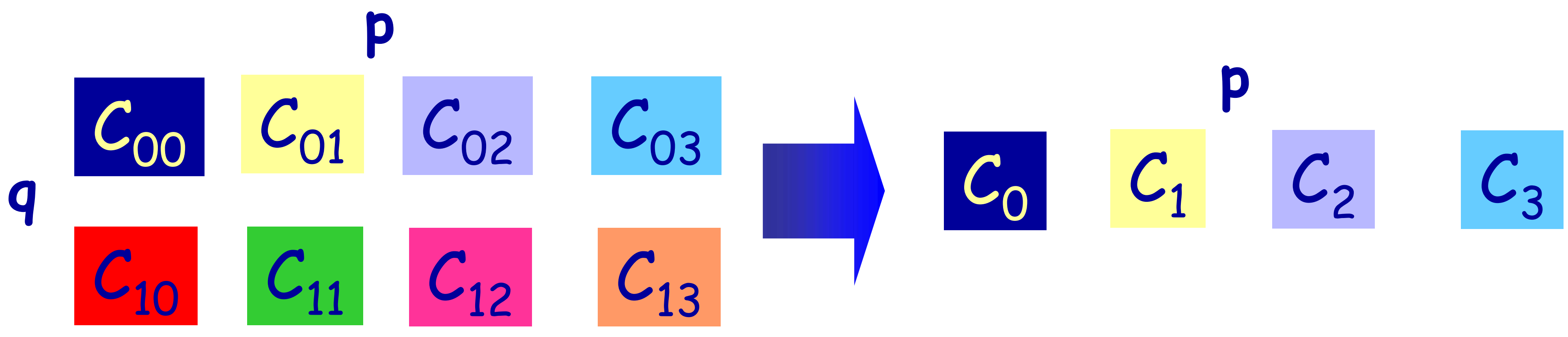


Speed-up/efficienza (**def classica**)

GRIGLIA qxp

Attenzione:

- è possibile riconoscere la I strategia dalla III quando p=1
- è possibile riconoscere la II strategia dalla III quando q=1



Speed-up/efficienza (**def classica**)

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

GRIGLIA qxp

Attenzione:

- è possibile riconoscere la I strategia dalla III quando p=1
- è possibile riconoscere la II strategia dalla III quando q=1
- cosa succede se **mod(N,q) ≠0** e/o **mod(M,p)≠0** ?

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

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

GRIGLIA qxp

- **Attenzione:**

  cosa succede se **mod(N,q) ≠0** e/o **mod(M,p)≠0** ?

Alcuni core si occuperanno di blocchi di matrice (e se serve del vettore) di dimensione maggiore

Es: RIGHE **mod(N,q)≠0**

**Il numero di righe che avanza
(cioè il resto della divisione)
viene assegnato a tutti i core riga che hanno
prima coordinata strettamente minore del resto**

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

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

GRIGLIA qxp

- **Attenzione:**

  cosa succede se **mod(N,q) ≠0** e/o **mod(M,p)≠0** ?

Alcuni processori avranno dei blocchi di matrice (e se serve del vettore) di dimensione maggiore

Es: RIGHE **mod(N,q)≠0**

I core che hanno coordinata riga strettamente minore del resto hanno una riga in più della matrice. Nessuna variazione invece per il blocco relativo al vettore!

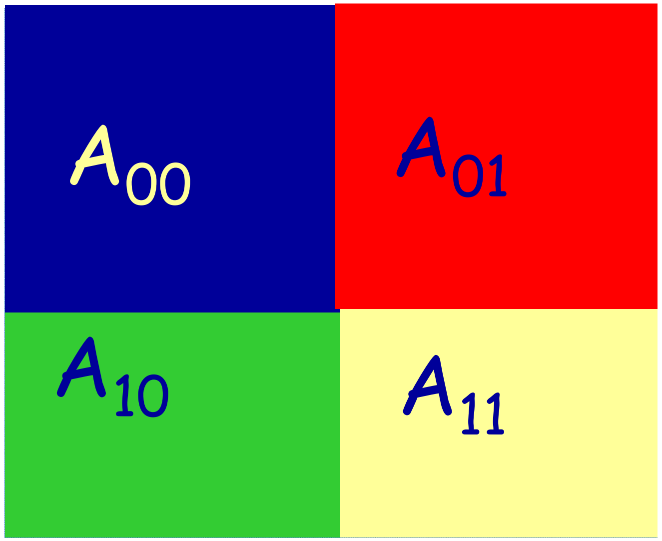
III Strategia: speed-up/efficienza (**def classica**)

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

Es: se **mod(N,q) ≠0**

GRIGLIA
qxp=4=2x2

A



$\dim[b_{loc}]= M/p$

$\dim[A_{10}]=(N/q)x(M/p)$

$\dim[A_{11}]=(N/q)x(M/p)$

$\dim[A_{00}]=(N/q + 1)x(M/p)$

$\dim[A_{01}]=(N/q + 1)x(M/p)$

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

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

Es: se **mod(N,q) ≠0**

GRIGLIA
qxp=4=2x2

$dim[A_{00}]=(N/q + 1)x(M/p)$

$dim[A_{01}]=(N/q + 1)x(M/p)$

$dim[b_{loc}]= M/p$

$dim[A_{10}]=(N/q)x(M/p)$

$dim[A_{11}]=(N/q)x(M/p)$

C_{00}	C_{01}
C_{10}	C_{11}

C_{00}: $dim[r_{00}]=(N/q + 1)$, C_{01}: $dim[s_{01}]=(N/q + 1)$

C_{10}: $dim[r_{10}]=(N/q)$, C_{11}: $dim[s_{11}]=(N/q)$

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

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

GRIGLIA qxp

I strategia per collezione vettori

$S_{qxp}(NxM) = T_1(NxM)/T_{qxp}(NxM) =$

$= N[2M-1] / (\boxed{[N/q+1]}\ [\ 2M/p-1\ ] + \boxed{(N/q+1)}\ (p-1))$

$S_{qxp}(NxM) = T_1(NxM)/T_{qxp}(NxM) =$

II strategia per collezione vettori

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

Non resta che calcolare l'overhead e fare osservazioni sull'isoefficienza.