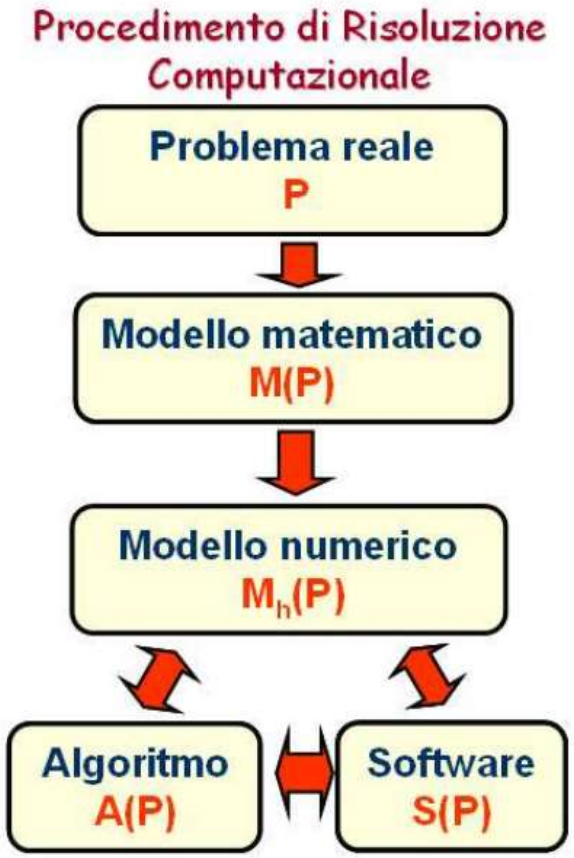
Calcolo Parallelo e Distribuito

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

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

Modellizzazione di problemi su larga scala

- Ricerca su internet
- Trasporto
- Pubblicità e Marketing
- Servizi bancari e finanziari
- Media e intrattenimento
- Meteorologia
- Assistenza sanitaria
- Sicurezza informatica
- Formazione



Il parallelismo delle architetture MIMD-SM

metodi numerici paralleli

Modello Numerico
$M_h(P)$

Ripartire dal modello numerico (modello matematico discretizzato **h=1,…,N**) e analizzare gli **N** passi in modo da distribuirli, eventualmente, a più unità processanti.
Più possibilità:

- **ogni unità** processante esegue un passo **differente**
  (**decomposizione funzionale**)
- **tutte le unità** processanti eseguono **la stessa** operazione su un sottoinsieme di dati
  (**decomposizione del dominio**)

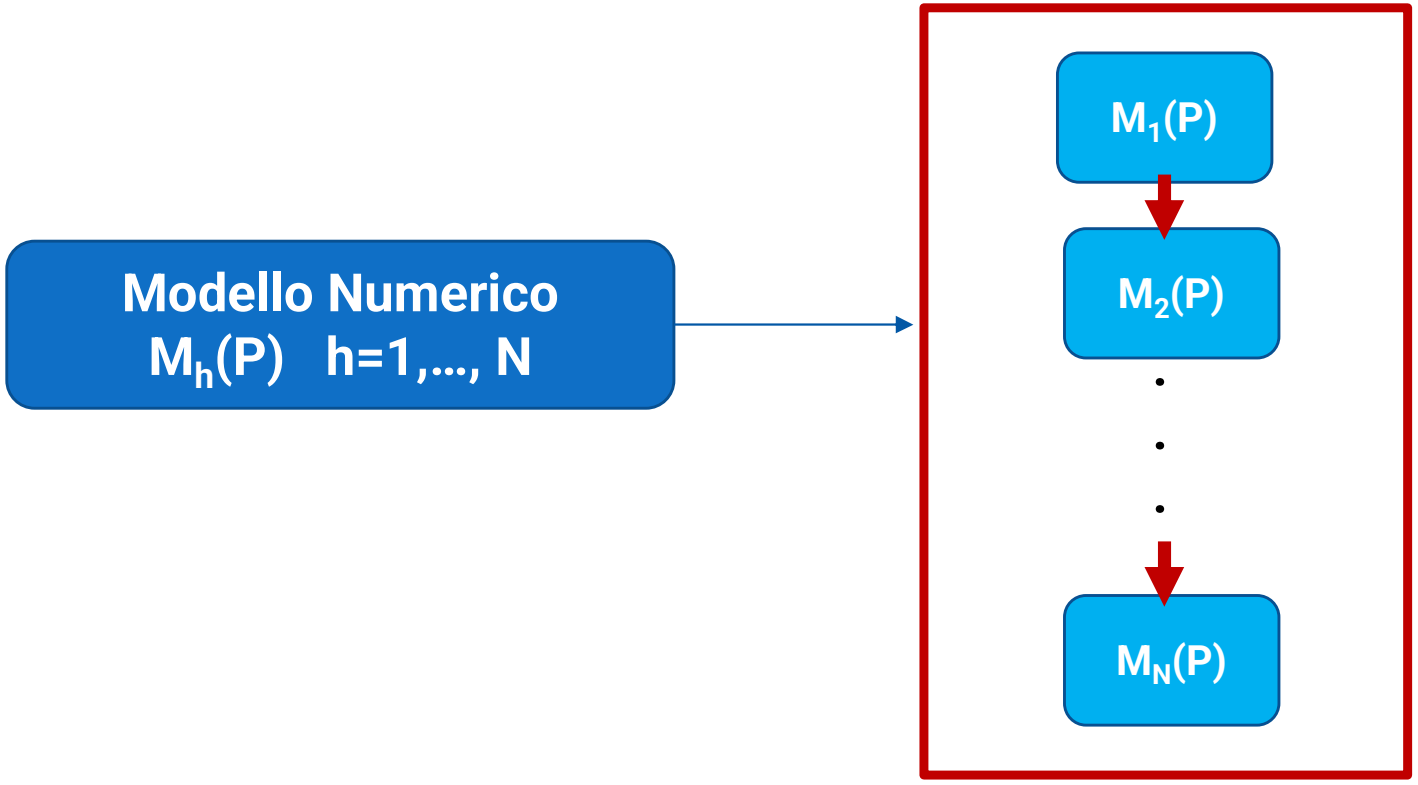
- **combinazione delle due possibilità precedenti**

Il parallelismo delle architetture MIMD-SM

metodi numerici paralleli

Modello Numerico
$M_h(P)$

Ripartire dal modello numerico (modello matematico discretizzato **h=1,…,N**) e **suddividere, quando possibile, i task in più sotto-task uguali e processarli contemporaneamente, ma riducendo al minimo la collezione dei risultati locali.**

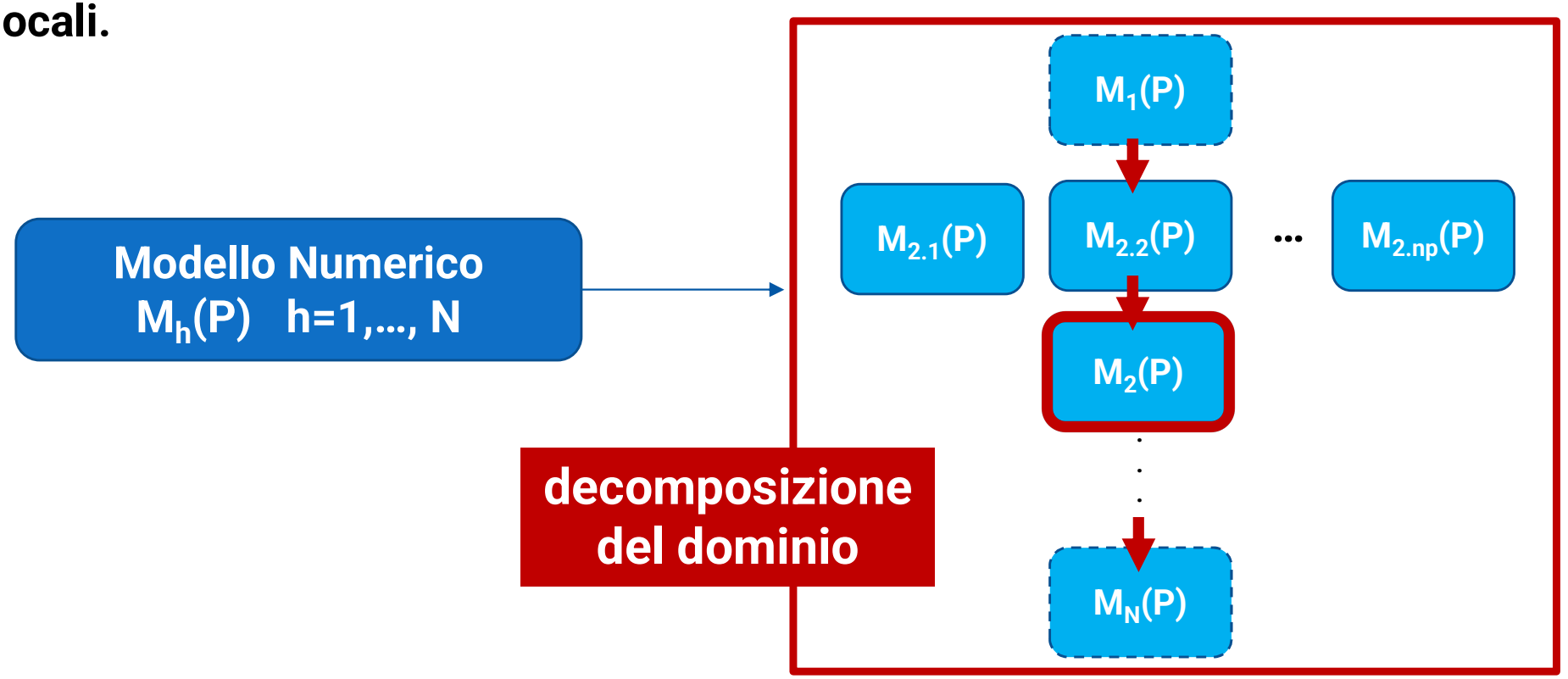


Il parallelismo delle architetture MIMD-SM

metodi numerici paralleli

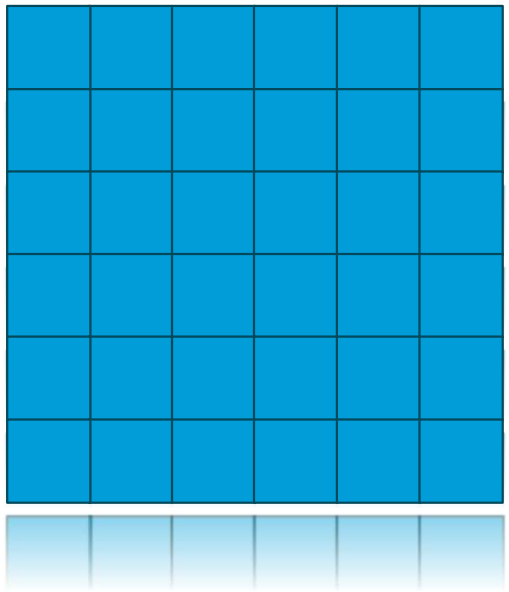
Modello Numerico
$M_h(P)$

Ripartire dal modello numerico (modello matematico discretizzato **h=1,…,N**) e **suddividere, quando possibile, i task in più sotto-task uguali e processarli contemporaneamente, ma riducendo al minimo la collezione dei risultati locali.**



Decomposizione di matrici

Prodotto di uno scalare con una matrice di grandi dimensioni!



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

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

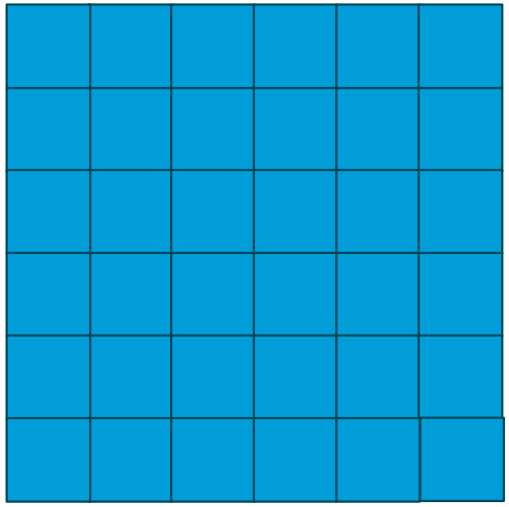
Prodotto di uno scalare con una matrice di grandi dimensioni!

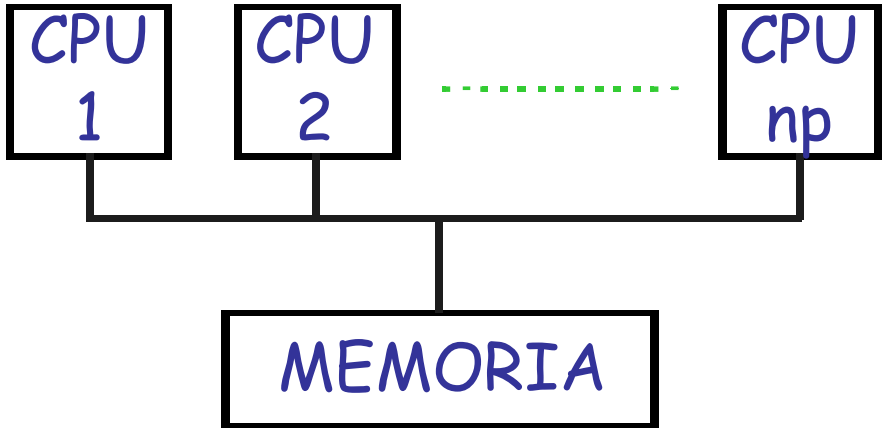
Su un calcolatore monoprocessore il prodotto è calcolato eseguendo *N×N prodotti uno per volta* secondo un ordine prestabilito

```
begin
     for i=0 to N-1 do
       for j=0 to N-1 do
             c_{i,j} := β·a_{i,j};
       endfor
     endfor
end
```

Decomposizione di matrici

Prodotto di uno scalare con una matrice di grandi dimensioni!





su un calcolatore parallelo

tipo MIMD

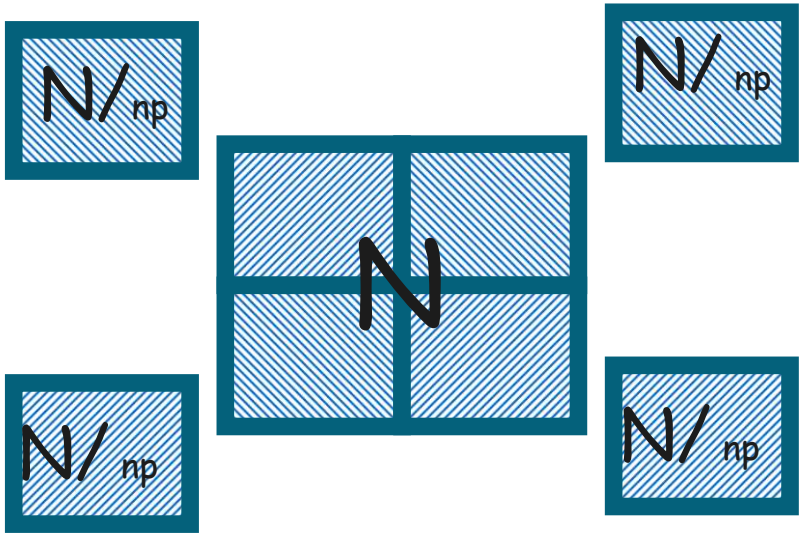
A MEMORIA **CONDIVISA**

Il parallelismo delle architetture MIMD

prodotto di uno scalare per una matrice di dimensione N×N

Se ho a disposizione np unità processanti, come posso procedere sfruttando il concetto di calcolo parallelo?

Decomporre un problema di dimensione N in **np** sottoproblemi di dimensione N/**np** e risolverli contemporaneamente usando **np** CPU



Quali sono i sotto-problemi indipendenti?

Decomposizione del problema

Scalare per Matrice

Partizionamento della matrice A

IN BLOCCHI

Riformulazione dell'algoritmo sequenziale

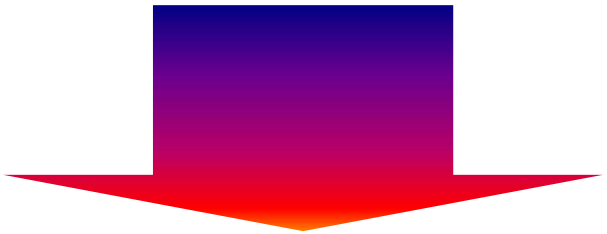
"A BLOCCHI"

Parallelismo dell'algoritmo

"A BLOCCHI"

Qual è l'algoritmo parallelo ?

Partizionamento della matrice



Distribuzione dei blocchi

ai processori

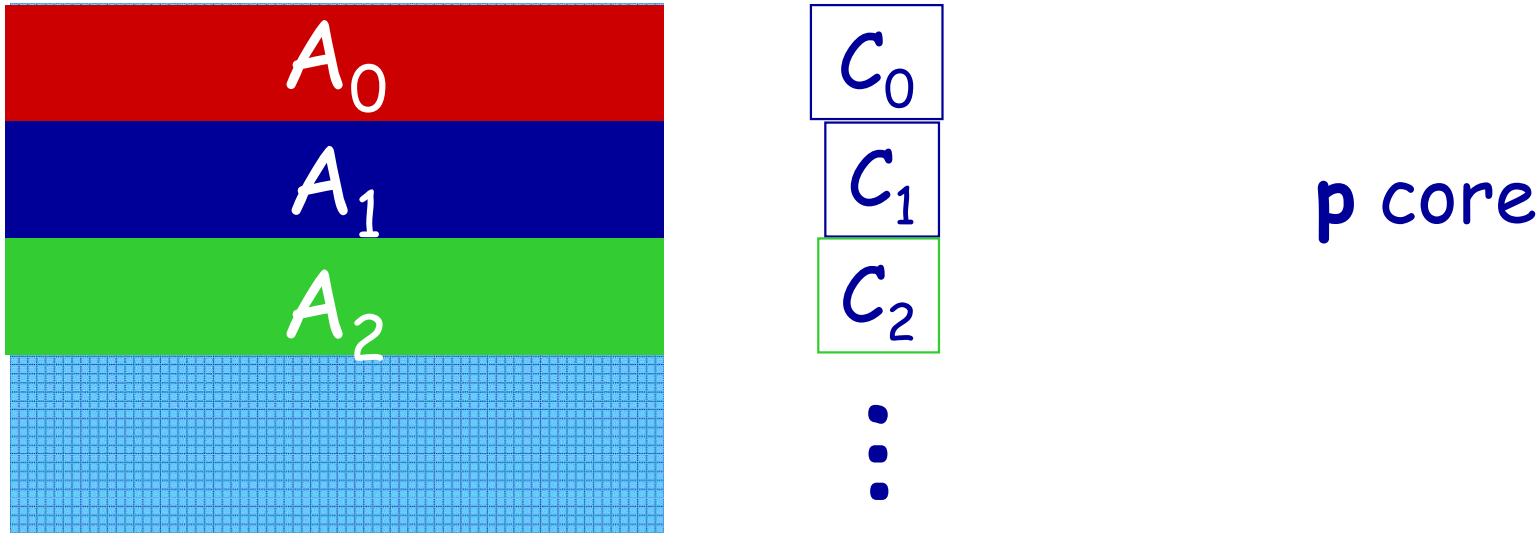
Algoritmo a blocchi

Algoritmo parallelo

I STRATEGIA

Suddividiamo la matrice A in

BLOCCHI di RIGHE



I strategia di parallelizzazione
prodotto di uno scalare per una matrice di dimensione N×N

IDEA

Suddividere il dominio del problema (la matrice) per blocchi di righe e assegnare il prodotto dello scalare per il **blocchi riga** ad ogni CPU

I blocchi riga calcolati possono essere uniti nella memoria shared per formare la matrice risultato

PARALLELISMO COMPLETO

Prodotto di uno scalare per una matrice
MIMD Shared Memory

I core possono accedere simultaneamente alla memoria globale su dati differenti

- Esempio: N=6, p=2

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix}$$

C_0

C_1

Prodotto di uno scalare per una matrice
MIMD Shared Memory

- Esempio: N=6, p=2

Prodotti locali

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \cdot \beta$$

c_0

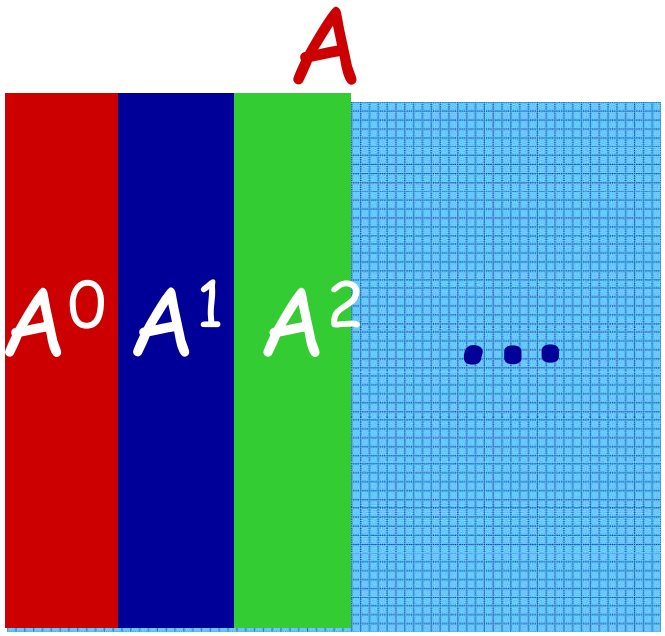
c_1

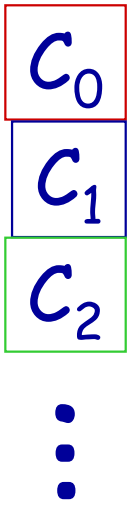
II STRATEGIA

Suddividiamo la matrice

A in

BLOCCHI di COLONNE





p core

II strategia di parallelizzazione
prodotto di uno scalare per una matrice di dimensione N×N

IDEA

Suddividere il dominio del problema (la matrice) per blocchi di colonne e assegnare il prodotto dello scalare per il **blocchi colonna** ad ogni CPU

I blocchi colonna calcolati possono essere uniti nella memoria shared a formare la matrice risultato

PARALLELISMO COMPLETO

Prodotto di uno scalare per una matrice
MIMD Shared Memory

I core possono accedere simultaneamente alla memoria globale su dati differenti

- Esempio: N=6, p=2

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix}$$

c_0 c_1

Prodotto di uno scalare per una matrice
MIMD Shared Memory

- Esempio: N=6, p=2

Prodotti locali

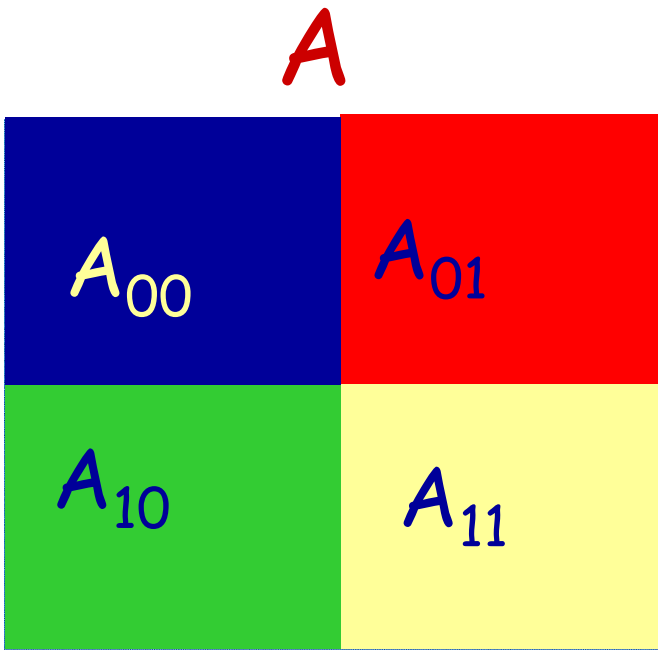
$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \cdot \beta$$

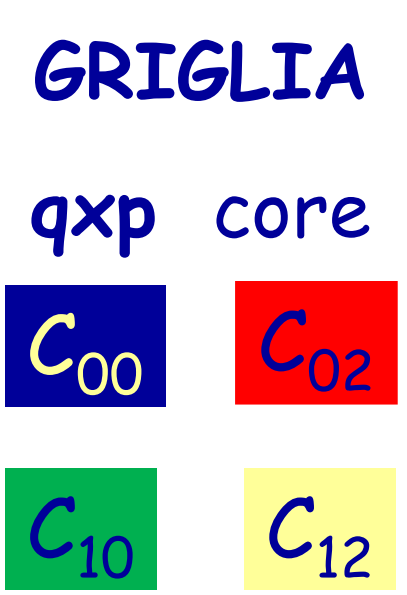
c_0 c_1

III STRATEGIA

Suddividiamo la matrice A in

BLOCCHI di RigheColonne





III strategia di parallelizzazione
prodotto di uno scalare per una matrice di dimensione N×N

IDEA

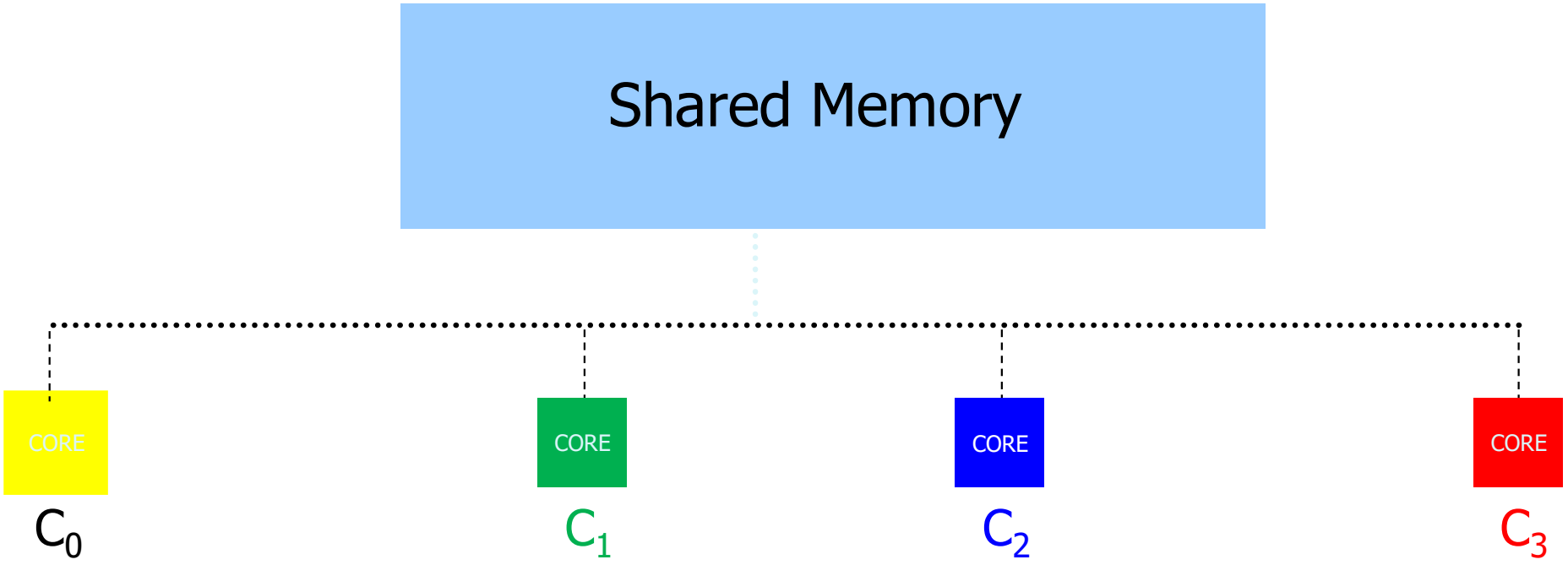
Suddividere il dominio del problema (la matrice) per blocchi di righe&colonne e assegnare il prodotto dello scalare per il **blocchi riga-colonna** ad ogni CPU

I blocchi riga-colonna calcolati possono essere uniti nella memoria shared la matrice risultato

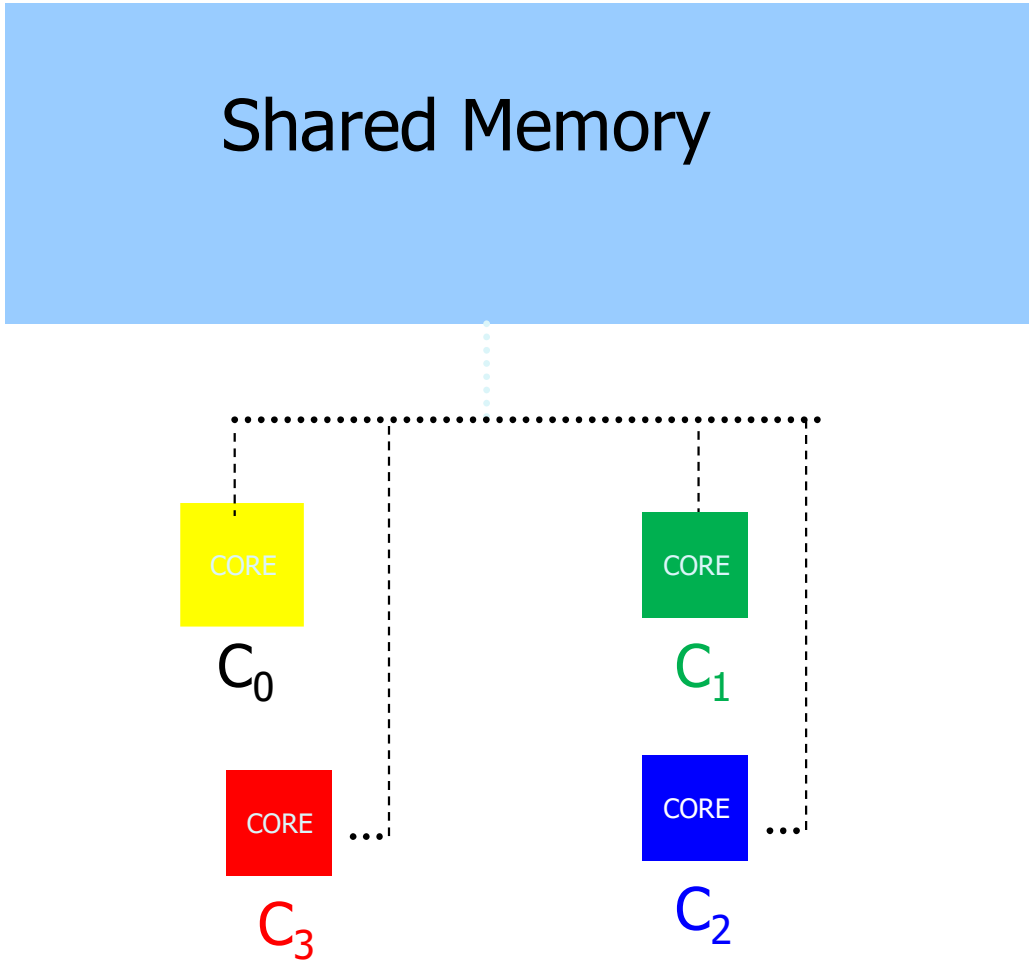
PARALLELISMO COMPLETO

Griglie di processori/core – TOPOLOGIA VIRTUALE

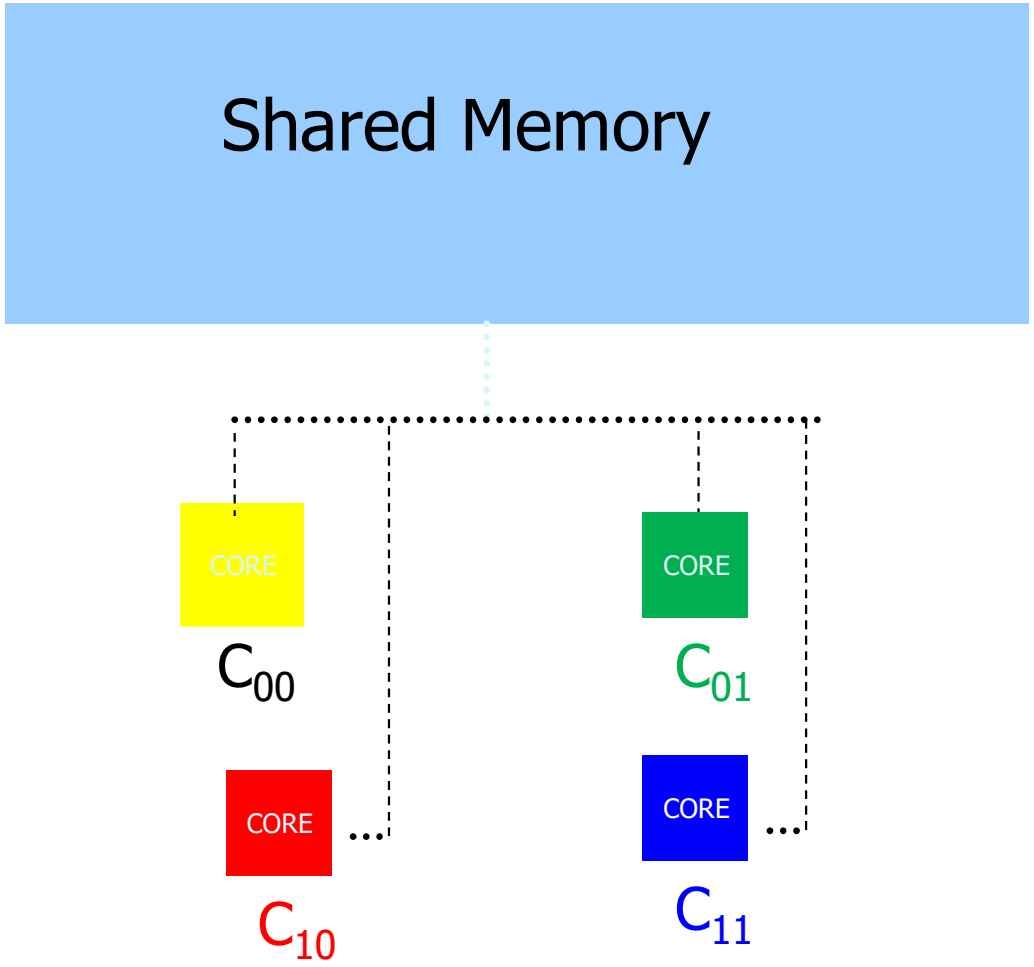
Griglie di processori/core
TOPOLOGIA VIRTUALE



Griglie di processori/core
TOPOLOGIA VIRTUALE



Griglie di processori/core
TOPOLOGIA VIRTUALE



Prodotto di uno scalare per una matrice
MIMD Shared Memory

I core possono accedere simultaneamente alla memoria globale su <u>dati differenti</u>

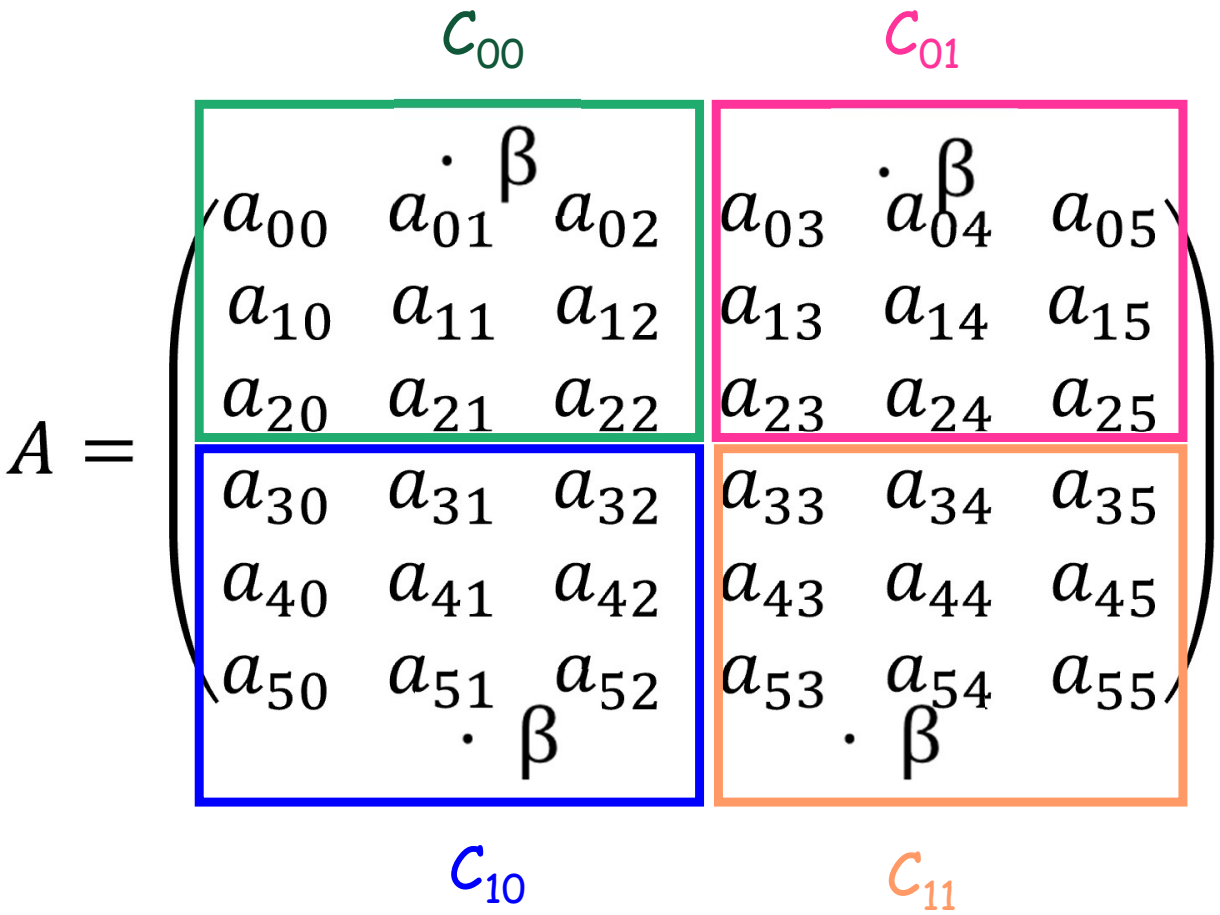
- Esempio: N=6, q×p=2×2

$$A = \begin{pmatrix} \overbrace{\begin{matrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{matrix}}^{c_{00}} & \overbrace{\begin{matrix} a_{03} & a_{04} & a_{05} \\ a_{13} & a_{14} & a_{15} \\ a_{23} & a_{24} & a_{25} \end{matrix}}^{c_{01}} \\ \underbrace{\begin{matrix} a_{30} & a_{31} & a_{32} \\ a_{40} & a_{41} & a_{42} \\ a_{50} & a_{51} & a_{52} \end{matrix}}_{c_{10}} & \underbrace{\begin{matrix} a_{33} & a_{34} & a_{35} \\ a_{43} & a_{44} & a_{45} \\ a_{53} & a_{54} & a_{55} \end{matrix}}_{c_{00}} \end{pmatrix}$$

Prodotto di uno scalare per una matrice
MIMD Shared Memory

- Esempio: N=6 , q×p=2×2

Prodotti locali



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 possible fare filtri base a qualunque tipo di segnale

Modellizzazione di problemi su larga scala

- Ricerca su internet
- Trasporto
- Pubblicità e Marketing
- Servizi bancari e finanziari
- Media e intrattenimento
- Meteorologia
- Assistenza sanitaria
- Sicurezza informatica
- Formazione