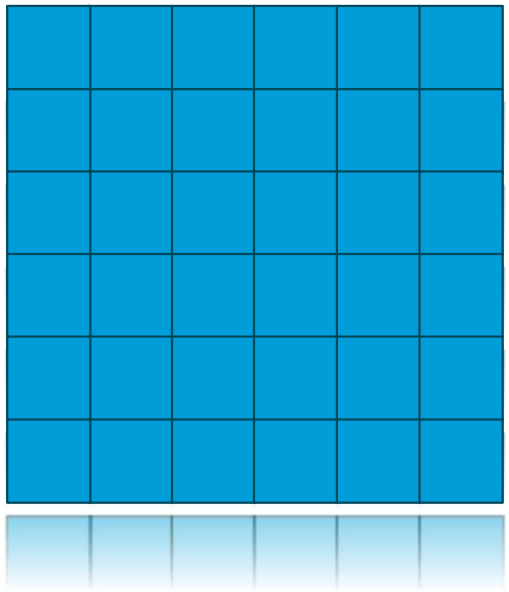
Calcolo Parallelo e Distribuito

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

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

Decomposizione di matrici

Prodotto di uno scalare con una matrice di grandi dimensioni!



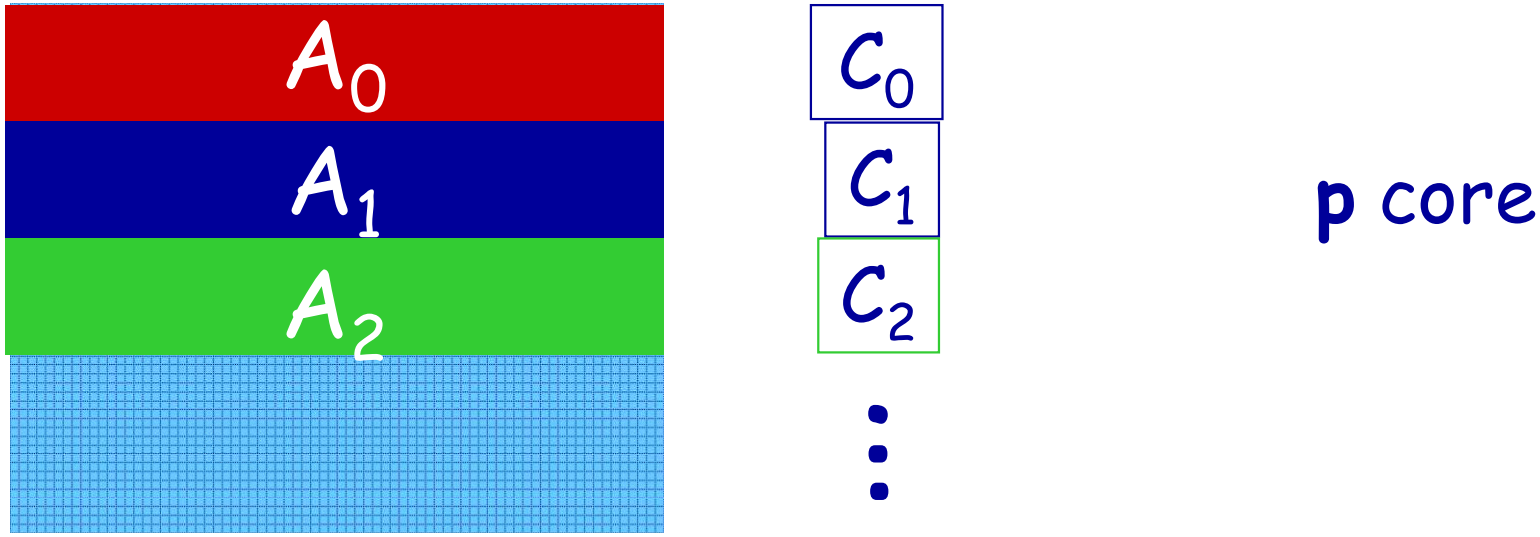
Input: $\beta \cdot A$: $\dim(A)=N \times N$

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

I STRATEGIA

Suddividiamo la matrice A in

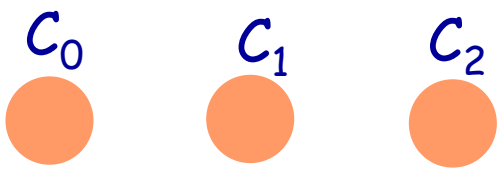
BLOCCHI di RIGHE

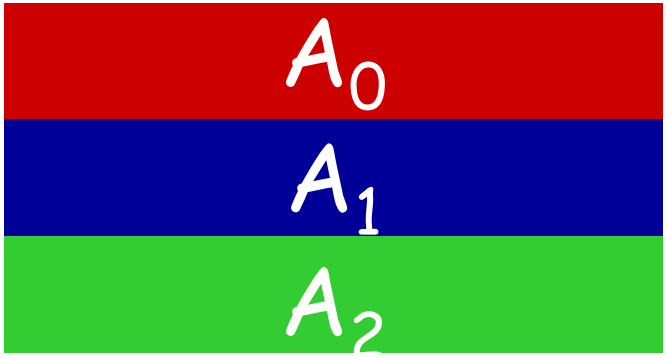


p core

p=3, dim[A]=N×N

prodotto di uno scalare per una matrice





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

In sequenziale

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

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

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

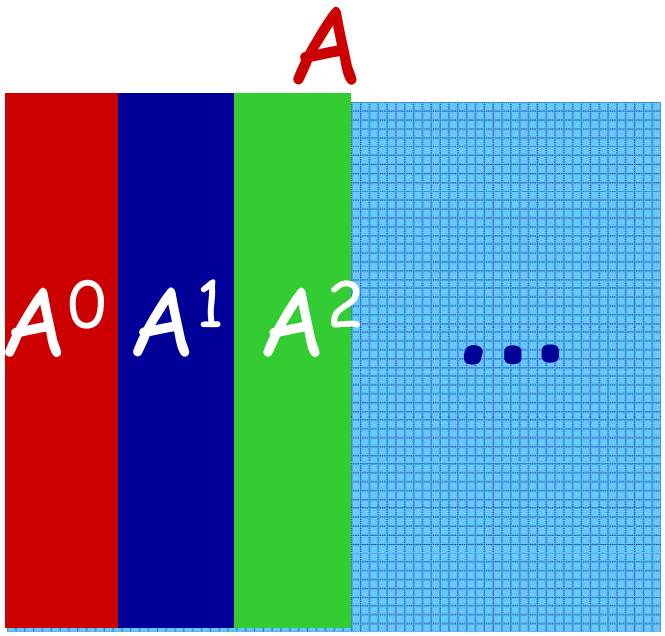
$$E_p = S_p / p = 1$$

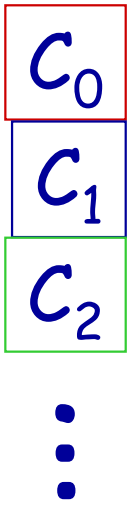
II STRATEGIA

Suddividiamo la matrice

A in

BLOCCHI di COLONNE

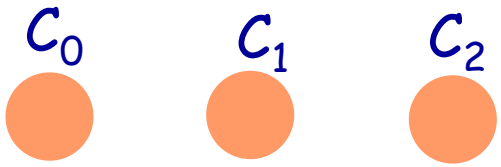


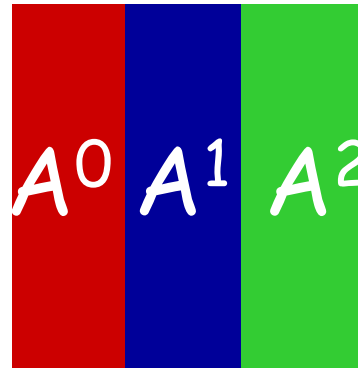


p core

prodotto di uno scalare per una matrice

p=3, dim[A]=N×N





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

In sequenziale

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

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

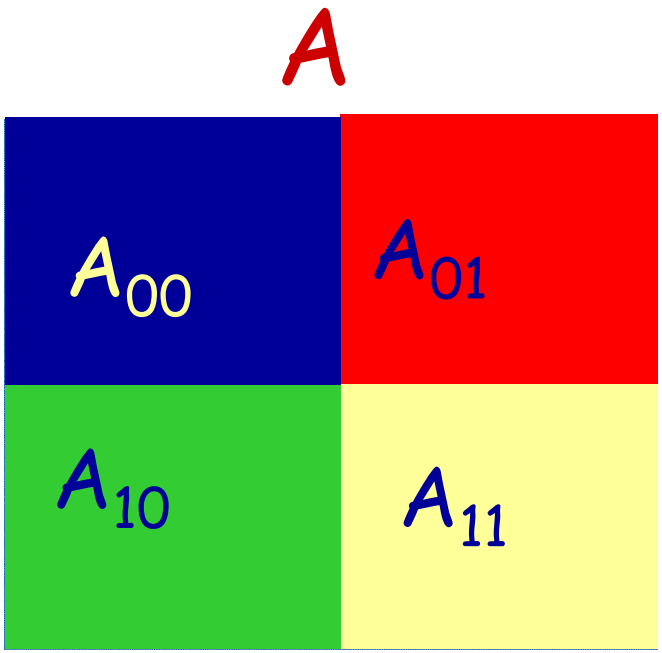
$$Oh = p\,T_p(N^2) - T_1(N^2) = p[N^2/p] - N^2 = 0$$

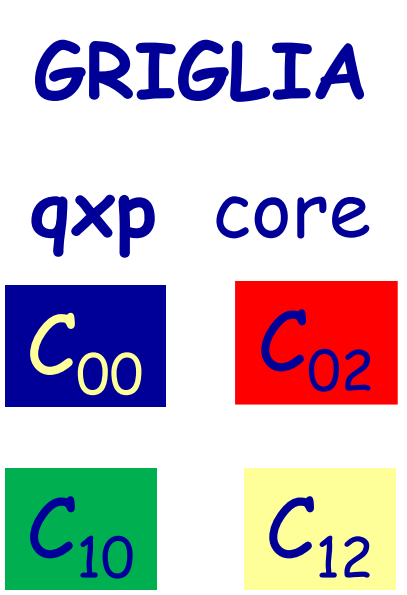
$$E_p = S_p/p = 1$$

III STRATEGIA

Suddividiamo la matrice A in

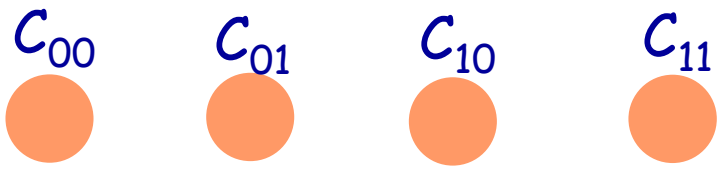
BLOCCHI di RigheColonne

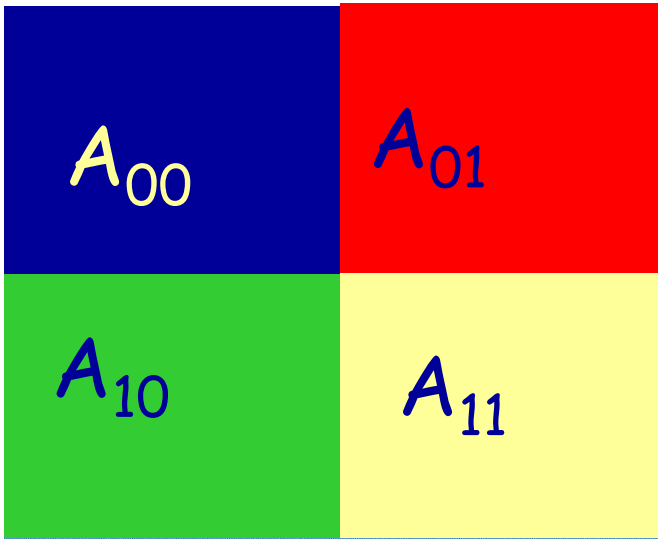




q×p= 2×2, dim[A]=N×N

prodotto di uno scalare per una matrice





3 strategia – q righe - p colonne

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

In sequenziale

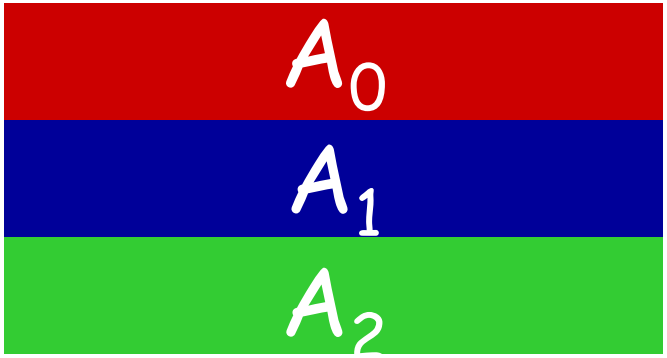
$$T_1(N^2) = N^2$$

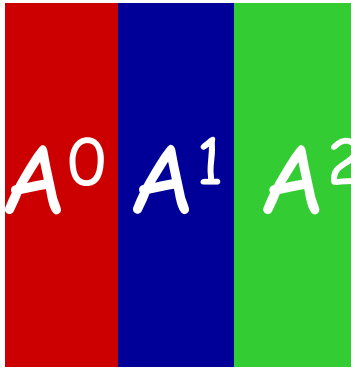
$$S_{qp} = T_1(N^2)/T_{qp}(N^2) = N^2/[(N/q)(N/p)] = qp$$

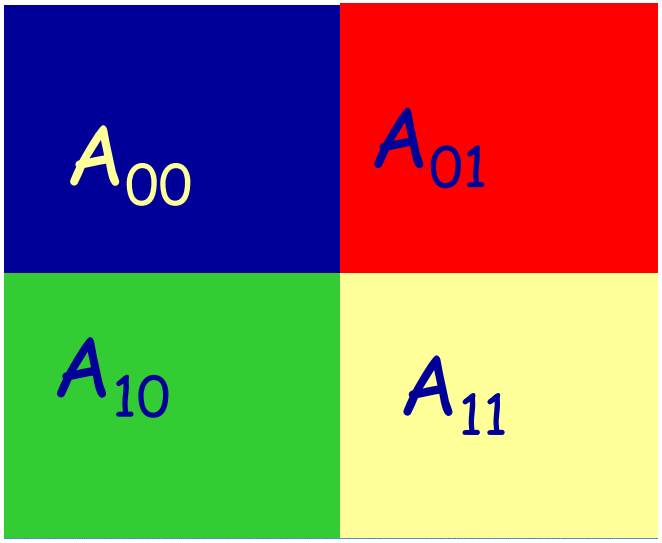
$$Oh = qp\ T_{qp}(N^2) - T_1(N^2) = qp[N^2/qp] - N^2 = 0$$

$$E_p = S_p/p = 1$$

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







Nucleo: Prodotto scalare per Matrice – 1 strategia

```
#pragma omp parallel for shared(m,n,A, B, alpha) private(i,j)
for (i=0; i<n; i++){
        for (j=0; j<m; j++)
                B[i][j] = A[i][j]*alpha;
}
```

Nucleo: Prodotto scalare per Matrice – 2 strategia

```
#pragma omp parallel for shared(m,n,A, B, alpha) private(i,j)
for (j=0; j<n; j++){
        for (i=0; i<m; i++)
                B[i][j] = A[i][j]*alpha;
}
```

Nucleo: Prodotto scalare per Matrice 3 strategia

Con la clausola collapse (2) è possibile espandere espandere il costrutto parallel for anche al secondo for

```
// Decomposizione per blocchi
#pragma omp parallel for collapse(2)
for (int blockRow = 0; blockRow < n; blockRow += blockSize) {
    for (int blockCol = 0; blockCol < m; blockCol += blockSize) {
        for (int i = blockRow; i < blockRow + blockSize && i < n; i++) {
            for (int j = blockCol; j < blockCol + blockSize && j < m; j++) {
                A[i][j] *= alpha;
            }
        }
    }
}
```