

**INDICAZIONI OPERATIVE PER LE
CONSEGNE DEGLI ELABORATI PER IL CORSO DI
CALCOLO PARALLELO E DISTRIBUITO MOD.B**

OPERAZIONI DA FARE 1 SOLA VOLTA

- Creare una cartella Dropbox con il nome NomeCognome (ad es. GennaroEsposito)
- Condividere la cartella con l'indirizzo email: marco.lapeгна@unina.it

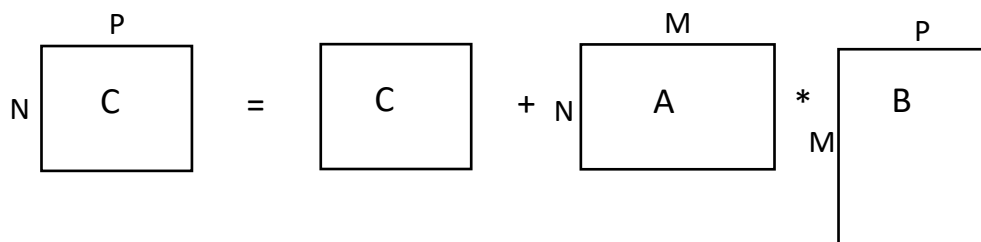
CASO DI STUDIO

date 3 matrici:

- A (N righe e M colonne)
- B (M righe e P colonne)
- C (N righe e P colonne)

gli elaborati sono finalizzati allo studio delle prestazioni di vari algoritmi per la risoluzione del problema

$$C = C + A * B \quad (*)$$



ELABORATI DA CONSEGNARE

1. matmat

Scrivere 6 function per la risoluzione del problema (*) con le seguenti testate

- o `void matmatikj(int LDA, int LDB, int LDC, double *A, double *B, double *C, int N, int M, int P)`
- o `void matmatkij(int LDA, int LDB, int LDC, double *A, double *B, double *C, int N, int M, int P)`

- o `void matmatijk(int LDA, int LDB, int LDC, double *A, double *B, double *C, int N, int M, int P)`
- o `void matmatjik(int LDA, int LDB, int LDC, double *A, double *B, double *C, int N, int M, int P)`
- o `void matmatkji(int LDA, int LDB, int LDC, double *A, double *B, double *C, int N, int M, int P)`
- o `void matmatjki(int LDA, int LDB, int LDC, double *A, double *B, double *C, int N, int M, int P)`

dove:

LDA, LDB, LDC : leading dimension delle matrici

*A, *B, *C : matrici

N, M, P : dimensioni delle matrici

Le 6 function sono ottenute dalle permutazioni degli indici delle strutture di iterazione alla base dell'algoritmo:

```

for ... = 0 to ...
  for ... = 0 to ...
    for ... = 0 to ...
      C(i,j) = C(i,j) + A(i,k) * B(k,j)
    endfor
  endfor
endfor

```

Valutare le prestazioni

$$Prestazioni = \frac{\text{tempo di esecuzione}}{\text{complessita' computazionale}}$$

delle 6 function per matrici quadrate (N = M = P) di ordine N = 50, 100, ... 1500 .
Determinare la function con le migliori prestazioni e la dimensione ottimale N_{opt} .

Inserire solo le funzioni assegnate in un unico file di nome `matmat.c` , il quale va poi inserito nella cartella Dropbox condivisa. Prestare particolare attenzione ai nomi dei file e ai nomi delle function !

2. matmatblock

Scrivere una function per la risoluzione del problema (*) mediante la versione a blocchi dell'algoritmo del prodotto di matrici, con la seguente testata:

```
o void matmatblock(int LDA, int LDB, int LDC, double *A,
    double *B, double *C, int N, int M, int P, int DBN, int
    DBM, int DBP)
```

dove:

LDA, LDB, LDC : leading dimension delle matrici

*A, *B, *C : matrici

N, M, P : dimensioni delle matrici

DBN, DBM, DBP : dimensione di blocchi relativi alle dimensioni N, M e P

La function `matmatblock` deve richiamare la function con le migliori prestazioni determinata nel punto 1. Come dimensione dei blocchi si usi $DBN = DBM = DBP = N_{opt}$ dove N_{opt} e' la dimensione ottimale determinata al punto 1.

Valutare le prestazioni della function `matmatblock`, confrontandole con quelle della migliore versione ottenuta nel punto 1, per matrici quadrate di ordine $N = M = P$ con $N = N_{opt}, 2*N_{opt}, \dots 10*N_{opt}$ (ad esempio, se nel punto 1 e' stato determinato $N_{opt} = 150$, eseguire i test con $N=150, 300, 450, \dots 1500$). Verificare che per la function `matmatblock` le prestazioni si mantengono costanti.

Inserire solo la funzione `matmatblock` in un unico file di nome `matmatblock.c`, il quale va poi inserito nella cartella Dropbox condivisa. Prestare particolare attenzione ai nomi dei file e ai nomi delle function !

3. `matmatthread`

Scrivere una function per la risoluzione del problema (*) mediante la versione multithread dell'algoritmo del prodotto di matrici, con la seguente testata:

```
o void matmatblock(int LDA, int LDB, int LDC, double *A,
    double *B, double *C, int N, int M, int P, int DBN, int
    DBM, int DBP, int NTrow, int NTcol)
```

dove:

LDA, LDB, LDC : leading dimension delle matrici

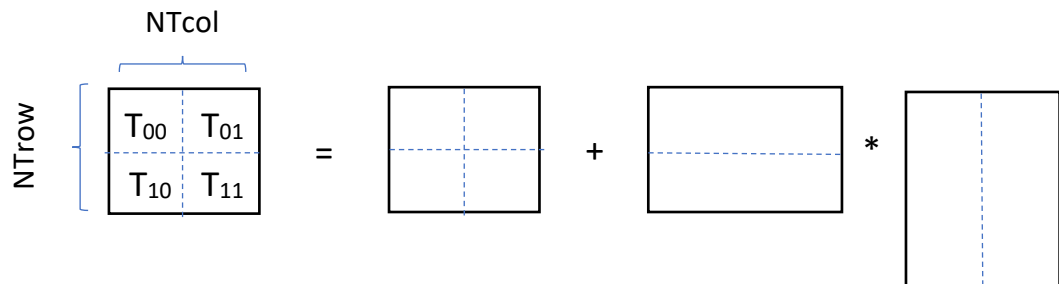
*A, *B, *C : matrici

N, M, P : dimensioni delle matrici

DBN, DBM, DBP : dimensione di blocchi relativi alle dimensioni N, M e P

NTrow, NTcol : Numero di thread lungo la direzione delle righe e delle colonne della matrice C (il prodotto NTrow*NTcol definisce il numero totale di thread da utilizzare).

Per la decomposizione del calcolo della matrice C tra i thread, si faccia riferimento alla seguente figura con 4 thread (NTrow = 2 , NTcol = 2)



La function `matmatthread` deve richiamare la function `matmatblock` sviluppata nel punto 2.

Valutare le prestazioni della function `matmatthread` utilizzando le seguenti configurazioni di thread:

NTrow = 1 e NTcol = 1

NTrow = 1 e NTcol = 2

NTrow = 2 e NTcol = 2

confrontandole con quelle della migliore versione ottenuta nel punto 1, e con quelle della funzione `matmatblock` sviluppata nel punto 2, con matrici quadrate di ordine $N = M = P$. Come dimensione delle matrici si usi N tale che esso sia multiplo di entrambi i prodotti $N_{opt} \cdot NTrow$ e $N_{opt} \cdot NTcol$ (ad esempio, se $N_{opt} = 150$ e $NTrow = NTcol = 2$, eseguire i test con $N=300, 600, 900 \dots 1500$). Valutare inoltre Speed-up ed Efficienza della funzione `matmatthread`.

Inserire la funzione `matmatthread` (ed eventuali altre function da essa richiamate, come ad esempio la funzione del thread) in un unico file di nome `matmatthread.c`, il quale va poi inserito nella cartella Dropbox condivisa. Prestare particolare attenzione ai nomi dei file e ai nomi delle function !