# Consegna

Eseguire la moltiplicazione tra due matrici su GPU sfruttando la shared memory.

Problema:

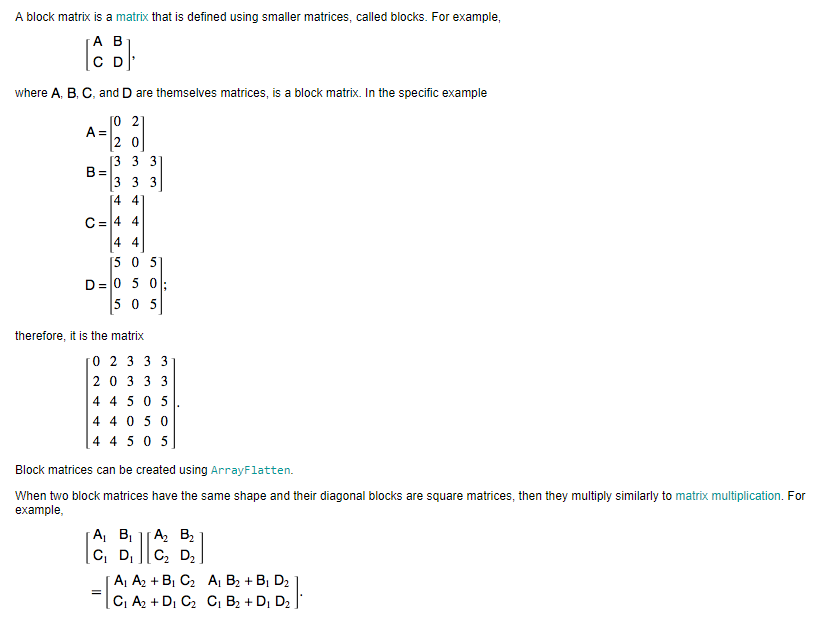
La shared memory per blocco, per la GPU considerata (Tesla K40), è di 49152 byte. Tale quantità di memoria non provoca particolari problematiche se la dimensioni delle matrici da moltiplicare è piccola, per esempio, se N = 10:

Ma se la dimensione inizia a diventare più importante, per esempio N = 100:

Valore che eccede la quantità massima di memoria shared utilizzabile in un blocco. Quindi non sarebbe possibile fare la moltiplicazione tra due matrici con tale dimensione in un singolo blocco.

Soluzione:

Per sfruttare l’elevato parallelismo all’interno di uno stesso blocco, senza eccedere la shared memory sfrutto la moltiplicazione per sottomatrici come spiegato nella figura sottostante.



In questo caso se la matrice è multipla di 32:

C00 = A00 \* B00 + A01 \* B10 + A02 \* B20

C01 = A00 \* B01 + A01 \* B11 + A02 \* B21

C10 = A10 \* B00 + A11 \* B10 + A12 \* B20

C11 = A10 \* B01 + A11 \* B11 + A12 \* B21

E così via..

Quindi posso spezzare l’operazione di moltiplicazione in 3 segmenti e salvare il risultato del blocco all’interno di una matrice “C locale”. Quindi per ogni blocco necessiterò di:

Valore che è al di sotto della massima quantità di memoria disponibile per blocco.

Compilazione