

Architetture e parallelismo

Esercizio 2

Nel file Esercizio2.c si può trovare un programma che calcola il prodotto tra tre matrici parallelizzando il calcolo sfruttando decomposizioni e thread.

Main

Nel main vengono settate prima di tutto le size delle matrici, se le dimensioni non vengono date da console durante la run allora avranno tutte e tre size 250x250. Una volta create le matrici la moltiplicazione tra esse viene delegata alla **triple_matrix_mul()**. Questa viene chiamata tante volte quanto è il numero massimo di thread settato (MAX_THREADS, settato da noi a 8).

triple_matrix_mul()

Si occupa della scomposizione delle matrici e delega la moltiplicazione dei blocchi per matrice ai vari thread. Ogni calcolo di $C*(A*B)$ viene eseguito 100 volte per avere una buona stima del tempo impiegato a fare il calcolo.

Osservazioni

Si può notare come al variare del numero di thread utilizzati e al variare della dimensione delle matrici varieranno anche le prestazioni del programma. In particolare, pensare che utilizzare più thread possa portare sempre a dei vantaggi tempistici è una considerazione ingenua, in quanto, ad esempio, nel caso in cui si abbiano matrici piccole andare a scomporle in blocchi da poche righe, se non una sola, assegnando ogni blocco ad un thread può portare ad un overhead dato dal context switch.

Il programma è consegnato con matrici 250x250 e 8 thread, che dà dei risultati molto buoni se si vuole dimostrare il miglioramento delle prestazioni tramite il calcolo parallelo. Si consiglia di provare il programma con matrici 4x4 e 4 thread per osservare il caso particolare spiegato sopra.