

Prodotto Matriciale

Ambiente di test

I test sono stati condotti su un Macbook Pro Santa Rosa equipaggiato con un processore **Intel Core duo T7500@2,2 Ghz** - Cache L2 4Mb.

- ♣ Il sistema operativo per i test è **Mac OS X 10.5**;
- ♣ i codici sorgente sono stati compilati con l'**i686-apple-darwin9-gcc-4.0.1**;
- ♣ per calcolare la durata del prodotto matriciale è stata utilizzata la funzione **clock()** all'interno della libreria **time.h** della libreria standard del C.

La potenza nominale del processore, secondo i dati forniti da Intel (<http://www.intel.com/support/processors/sb/cs-023143.htm>) è di **17,6 GFlops**.

I test sono stati eseguiti utilizzando un input prima a singola, poi a doppia, precisione.

L'input di riferimento è quello della matrice 512x512; a partire da questa sono state costruite le stime dei tempi di esecuzione delle matrici di dimensione maggiore.

Una curiosità forse ovvia per molti (per me non lo era): effettuando i test alimentando l'elaboratore a batteria si hanno prestazioni un po' inferiori molto probabilmente dovute al risparmio energetico.

Alcuni dati

Come presentato a lezione, il prodotto tra due matrici ha complessità $O(2n^3 - n^2)$; di conseguenza il numero di operazioni necessarie per eseguire il prodotto è dato dalla seguente tabella :

512	268 Mflop
1024	2146 Mflop
2048	17175 Mflop
4096	137422 Mflop

(la parte sinistra rappresenta la dimensione della riga/colonna della matrice, la parte destra il numero di operazioni totali necessarie)

Dividendo il numero di operazioni floating-point necessarie al calcolo, per il tempo di esecuzione si ottiene la quantità di operazioni floating-point eseguite per unità di tempo.

Prodotto con Metodo classico

Opzione “-fast” del gcc: ottimizza la velocità del codice (APPLE ONLY)

SINGOLA PRECISIONE

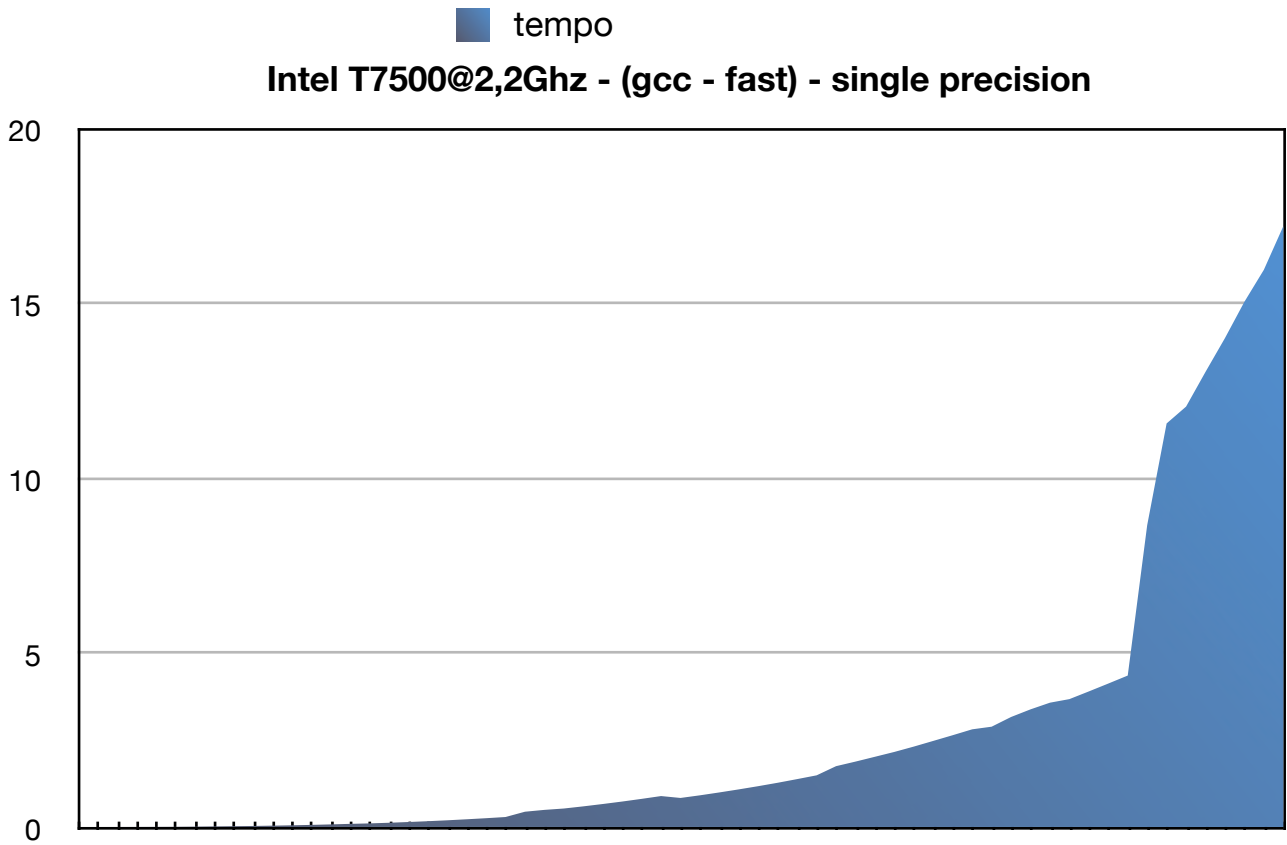
Dimensione Input	Tempo Atteso	Tempo Effettivo	MFlops
512x512	-	1,1sec	243
1024x1024	8,8 sec	15,3 sec	140
2048x2048	70,7 sec	203,2 sec	84,5

DOPPIA PRECISIONE

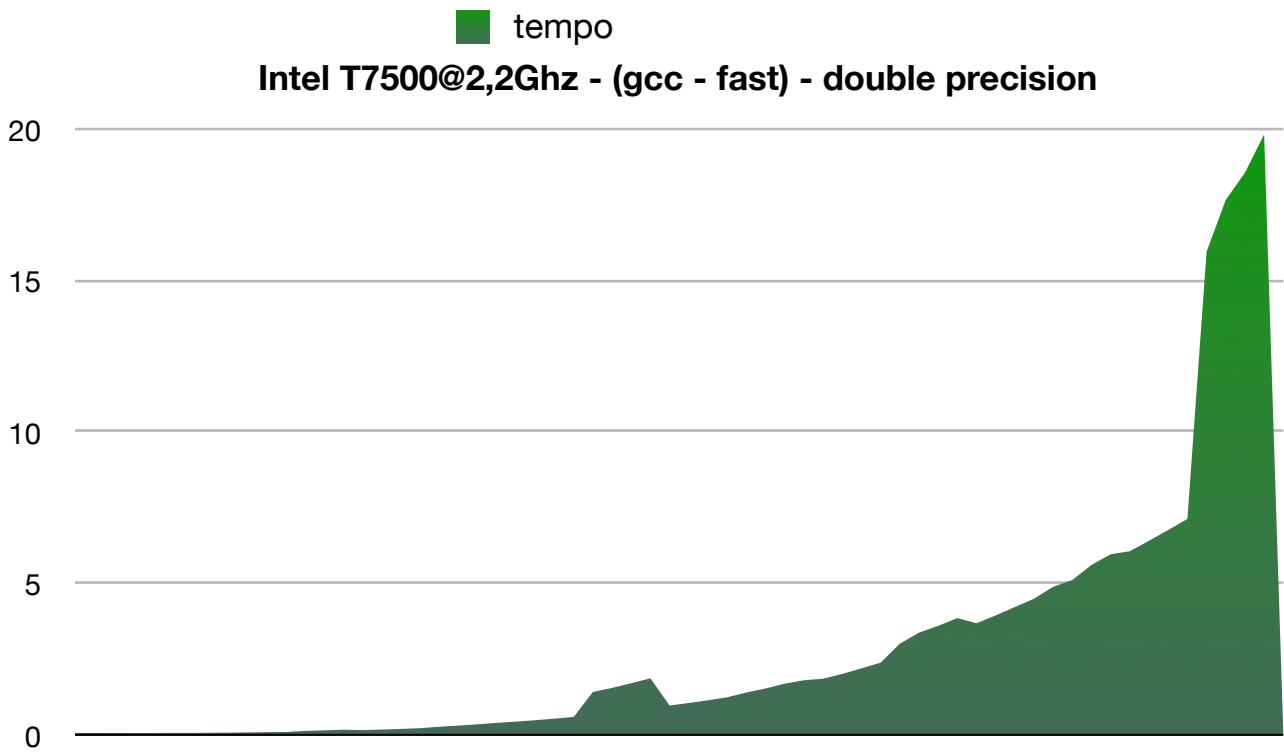
Dimensione Input	Tempo Atteso	Tempo Effettivo	MFlops
512x512	-	1,97 sec	136
1024x1024	15,77 sec	20,29 sec	105
2048x2048	126 sec	215 sec	79,7

Dai test effettuati sembrerebbe che le prestazioni scalino decisamente meglio quando l'input è formato da variabili a doppia precisione. In particolare se osserviamo il rapporto tra gli MFlops per la matrice 512x512, è di quasi 2 a 1 a favore della singola precisione. Tuttavia già per un input 2048x2048, ci accorgiamo che il divario fra gli Mflops nei due casi è decisamente diminuito.

GRAFICO PRESTAZIONI



Il grafico rappresenta il tempo di esecuzione al crescere dell'input (da 16x16 a 1024x1024); si nota una crescita improvvisa quando l'input supera la dimensione 850x850 (il dataset del programma non entra più in cache?).



Anche nel caso della doppia precisione non si ha una crescita lineare dei tempi di esecuzione; inoltre in alcuni casi accade che un input di dimensioni maggiori richieda un minor tempo di esecuzione. La cosa si è verificata in tutte le prove effettuate, quindi è difficilmente imputabile a fattori esterni. Anche in questo caso la dimensione dell'input parte da matrici 16x16 fino a matrici 1024x1024; anche in questo caso si verifica una crescita improvvisa del tempo di esecuzione quando si superano le dimensioni 960x960 dell'input (in particolare si passa da 7,08 secondi a 15,90 secondi a fronte di una crescita dell'input di 16 righe e 16 colonne).

TODO: Restano da effettuare i test con altre opzioni di compilazione e con l'altro algoritmo per il calcolo del prodotto proposto a lezione.