

# Algoritmi per il Calcolo Parallelo

Marco Tieghi

## Richiesta

L'esercizio richiedeva la realizzazione di un programma in grado di effettuare una fattorizzazione QR ricorrendo all'algoritmo di Gram-Schmidt modificato, usando prima un approccio in seriale (scrivendo cioè un programma che possa eseguire in seriale il codice), poi un approccio in parallelo (scrivendo un programma in grado di ripartire i calcoli su più unità di calcolo in contemporanea). Per l'approccio in parallelo si richiedeva l'uso degli strumenti messi a disposizione dalle librerie di CUDA, il sistema di calcolo tramite GPU sviluppato da NVIDIA.

## Approccio

Prima di tutto si è ideato e sviluppato il programma in seriale, in quanto ha permesso di comprendere meglio che algoritmo si doveva applicare. Si è scritto il codice in C, usando gli strumenti di questo linguaggio per strutturare il programma, e si sono definite le funzioni secondo il prototipo fornito con il testo dell'esercitazione. Da queste basi si è potuto scrivere così il programma in seriale.

In seguito si è cercato di comprendere meglio i meccanismi di CUDA esposti a lezione, cercando di capire la struttura di un programma CUDA e l'approccio verso questo sistema.

Compresi perciò i meccanismi, si è potuto modificare il codice in seriale per adattarlo alla struttura CUDA, modificando le funzioni in modo tale che potessero lavorare con i threads, definendo i diversi kernel del programma e calcolando la giusta struttura di blocchi e threads che si sarebbero usati, anche seguendo le indicazioni della consegna.

Nel codice ogni funzione e ogni passo è descritto così da avere ben chiaro cosa si sta eseguendo.

## Risultati

Il programma deve lavorare considerando una matrice  $A$   $m \times n$ , per la quale sono date alcune dimensioni di test: in un primo test la matrice è  $400 \times 300$ , in secondo test la matrice è  $1000 \times 800$ .

Si sono eseguiti tutti i test in locale, su un computer dotato di GPU NVIDIA GeForce GT 750M con 2GB di memoria DDR5, su cui sono installati i tool-kit CUDA.

### Matrice $400 \times 300$

Queste sono le diverse metriche ottenute per il tempo di calcolo della fattorizzazione usando una matrice  $400 \times 300$

- Tempo CPU codice seriale: 0.14118 s
- Tempo GPU codice parallelo: 0.33932 s
- Bandwidth GPU: 3.53648 GB/s
- Speedup: tempo CPU codice seriale/Tempo GPU codice parallelo  
= 0.42

### Matrice $1000 \times 800$

Queste sono le diverse metriche ottenute per il tempo di calcolo della fattorizzazione usando una matrice  $1000 \times 800$

- Tempo CPU codice seriale: 6.18393 s

- Tempo GPU codice parallelo: 3.20528 s
- Bandwidth GPU: 2.49588 GB/s
- Speedup: tempo CPU codice seriale/Tempo GPU codice parallelo  
= 1.929