

Domaći zadatak

25. maj 2019.

1. Data je realna matrica A dimenzija $n \times n$ i realni koeficijenti $c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9$. Transformisati svaki element matrice $A_{r,c}$, u vrsti r i koloni c , na sledeći način:

$$A'_{r,c} = c_1 \cdot A_{r-1,c-1} + c_2 \cdot A_{r-1,c} + c_3 \cdot A_{r-1,c+1} + c_4 \cdot A_{r,c-1} + c_5 \cdot A_{r,c} + c_6 \cdot A_{r,c+1} + c_7 \cdot A_{r+1,c-1} + c_8 \cdot A_{r+1,c} + c_9 \cdot A_{r+1,c+1}$$

Ukoliko neki od elemenata $A_{i,j}$ leži van matrice, smatrati da ima vrednost 0 u navedenoj formuli. Napisati kernel koji računa matricu A' .

2. (*problem ranca*) Dato je n objekata čije su zapremine celobrojne z_1, z_2, \dots, z_n i njihove odgovarajuće cene celobrojne vrednosti c_1, c_2, \dots, c_n , tj. objekat zapremine z_i ima cenu c_i . Potrebno je u kutiju zapremine Z , koja je ceo broj, upakovati objekte sa indeksima i_1, i_2, \dots, i_k tako da im je zbir zapremina manji ili jednak Z (tj. $z_{i_1} + z_{i_2} + \dots + z_{i_k} \leq Z$) a ukupna cena najveća moguća (najveće $c_{i_1} + c_{i_2} + \dots + c_{i_k}$).

3. Napisati kernel koji paralelno određuje da li je broj prost ili ne.

3*. Napisati kernel za Eratostenovo sito koje generiše niz p sastavljen od prvih k prostih brojeva. Nakon toga napisati kernel koji ispituje da li je broj prost koristeći niz p .

4. Napisati kernel koji množi dve matrice.

Napomene:

- za svaki zadatak napisati funkciju koja će da se izvršava na *host-u* (CPU) i koja se koristi da se ispita da li je vrednost koja je izračunata na GPU tačna
- napisati nekoliko varijanti kernela: prvo bez ikakve optimizacije korišćenja memorije, a nakon toga verziju gde se efikasno koristi deljena (*shared*) memorija, koristi se efikasno čitanje, odnosno, pisanje u globalnu memoriju, kao i konstantna memorija.
- tamo gde je moguće, napisati i varijantu kernela gde se vrši deljenje vrednosti u registrima kod niti (thread-ova) u jednom warp-u, tj. *warp shuffle*
- uporediti brzinu izvršavanja, brzinu protoka memorije kao i broj transakcija po zahtevu za različite verzije kernela
- kod drugog zadatka problem rešiti putem dinamičkog programiranja:
 1. konstruisati matricu $d[k, v]$ gde se računa najveća moguća postignuta cena objekata koji imaju u zbiru zapreminu v i koristi se prvih k objekata
 2. na početku se inicijalizuje cela matrica d na vrednost 0, i za prvi objekat imamo da je $d[1, z_1] = c_1$
 3. onda za svako $k > 1$ važi:

$$d[k, v] = \begin{cases} d[k-1, v] & \text{ako je } z_k > v \\ \max\{d[k-1, v], d[k-1, v-z_k] + c_k\} & \text{ako je } z_k \leq v \end{cases}$$

4. matrica d je dimenzija $n \times V$, ali za računanje k -tog reda potreban nam je samo $(k-1)$ -vi red, tako da je dovoljno koristiti dva niza matrice.