CS6023: GPU Programming

Assignment 3 (13 marks)

Submission deadline: March 15, 2020, 23:55 on Moodle

1 Problem specification

You are given a small database and a fixed number of update queries as input. You are required to simulate the execution of these queries ***in parallel*** on the database using a *GPU*. Maintaining the consistency of the database throughout is a primary goal. At the end of the simulation, the complete database should be written to a file.

The database:
For simplicity, in each test case, our database will contain only one table that has 'm' rows and 'n' attributes specified in the order C_1,C_2,...,C_n, where C_1 will always be the prime attribute (primary key). The first two attributes(C_1,C_2) will be read only attributes.

The queries:
There will be a fixed number (q) of update queries in each test case. The format of each query will be as follows:

$$\text{U } C_i \ X_i \text{ p } C_a \ X_a + C_b \ X_b + C_c \ X_c \text{ - } \cdots \ C_p \ X_p +$$

- The above query is to perform **p** update operations each on all the rows that satisfy the condition $C_i = X_i$.
- The **p** update operations will be: $C_a = C_a + X_a$, $C_b = C_b + X_b$, $C_c = C_c$ - X_c, $\cdots$ $C_p = C_p + X_p$
- The attribute used in update condition for all the queries is either C_1 or C_2 as they are read-only attributes.
- Only addition and subtraction operations (relative to the current value of an element) will be supported by the queries.

Input Format:

Your code should take the name of the input file from which it has to read the input and the name of the output file to which it has to write the output as command line arguments. For example, you should be executing your program like this:
./a.out input.txt output.txt

- The first line of the input file contains two integers 'm' and 'n', the number of rows and the number of columns in the table, respectively.
- Each of the next 'm' lines contains 'n' integers. Each such line contains the values of a row in the database.
- The next line contains a single integer 'q', denoting the number of the queries.
- The next 'q' lines contain the queries in the format specified above.

Output Format:
The output is defined as the final state of the database ('m' rows and 'n' columns), after performing all the update queries on it. Each line in the output file should contain one complete row of the database.

Constraints:
$1 \leq m \leq 2000$
$1 \leq n \leq 20$
$1 \leq q \leq 1500$
$1 \leq p \leq 20$

Sample Input:
5 4
1 3 209 150
2 4 283 241
3 4 172 29
4 5 121 264
5 4 184 190
4
U C2 5 2 C4 29 - C4 23 +
U C2 3 2 C4 17 + C3 12 +
U C2 4 2 C4 45 + C4 29 -
U C1 5 1 C4 9 +

Sample Output:
1 3 221 167
2 4 283 257
3 4 172 45
4 5 121 258
5 4 184 215

2 Submission guidelines

- Create a directory with your ROLL_NUMBER as the name and keep all the program files you want to submit inside it, and then compress it into a tar ball.
- Submit only one tar ball (.tar.gz) that contains your implementation on moodle: https://courses.iitm.ac.in/mod/assign/view.php?id=41722
- The name of the file submitted should strictly be of the format ROLL_NUMBER.tar.gz

  For example, if your roll number is CS16D019, the name of the file you submit should be CS16D019.tar.gz
- Make sure that the ROLL_NUMBER part of the filename is in upper case.
- Do not upload anything other than the ROLL_NUMBER.tar.gz file.
- After submission, download the file and make sure it was the one you intended to submit.

Learning suggestions:

- Write a CPU-version of code achieving the same functionality. Time the CPU code and GPU code separately for a large database and a large number of queries and compare the performances. You can also verify the correctness of your parallel program by comparing the results to that of the CPU version.