

Master 1^{ère} année

Dév. GPGPU

Examen — novembre 2019

Durée: 1h30 — Documents autorisés

1 – Consider the following procedure:

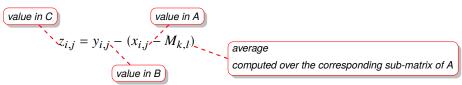
10pts ● we get two arrays A and B of dimensions 1024x2048;

- 2 each array contains temperature measures obtained during an experiment in plasma fluid theory;
- **3** we devide the array A into sub-matrices of 8x8, located by k, l in A:

$x_{0,0}$	$x_{0,1}$	$x_{0,2}$	$x_{0,3}$	$x_{0,4}$	$x_{0,5}$	$x_{0,6}$	$x_{0,7}$
$x_{1,0}$	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$	$x_{1,7}$
$x_{2,0}$	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$	$x_{2,7}$
$x_{3,0}$	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$x_{3,6}$	$x_{3,7}$
$x_{4,0}$	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$	$x_{4,5}$	$x_{4,6}$	$x_{4,7}$
$x_{5,0}$	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	x _{5,4}	x _{5,5}	x _{5,6}	x _{5,7}
$x_{6,0}$	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$	<i>x</i> _{6,5}	$x_{6,6}$	<i>x</i> _{6,7}
x _{7,0}	x _{7,1}	x _{7,2}	x _{7,3}	x _{7,4}	x _{7,5}	x _{7,6}	x _{7,7}

where:

- \diamond k varies from 0 to 127;
- \diamond *l* varies de 0 to 255;
- for each sub-matrix of A, we compute the arithmetic mean $M_{k,l} = \frac{\sum_{a=0}^{7} \sum_{b=0}^{7} x_{a,b}}{64}$;
- we fill the array C of dimensions 1024x2048 according to:
 - \diamond each value $z_{i,j}$ of C is equal to:



where i varies from 0 to 1023 and j varies from 0 to 2047.

Questions:

- a. How much **arithmetic means of sub-matrices** of A will we compute? (1pt)
- b. Give a configuration for **grid**, **block**, **thread** to solve the problem. (1pt)
- c. To compute $M_{k,l}$, could we use the **shared memory**? (2pts) Will it be **interesting**?

How to proceed?

- d. Write the **CUDA program** giving the desired result. (4pts)
- e. If we want to process a new array B without changing the array A, how could we proceed with the **best** (2pts) result?

You will write the new code to add at your program.



2 – Consider the following program:

10pts

```
1 #include <stdio.h>
 2
  #include <cuda.h>
 3
     removed code */
 4
   __global__ void my_kernel(float *t, int r)
 6
    int position = threadIdx.x + r;
 7
 8
9
    if ((position %2) == 0)
10
11
       t[position/2] = t[position] * t[position+1];
12
13 }
14 int main (void)
15 {
16
    float *gpu_data;
   int t = SIZE;
17
18 /* removed code */
   cudaMemcpy(gpu_data, data, 2*SIZE*sizeof(float), cudaMemcpyHostToDevice);
19
20
    while(t>1)
21
22
       my_kernel<<<1, SIZE>>> (gpu_data, t);
23
      t=t/2;
24
  /* removed code */
```

Code has been removed in lines 4, 18 and 25.

Questions:

a. Describe and explain what the program do.

b. Are the memory accesses performed optimally?
Explain your answer.

c. Is the program easily scalable: could we expand TAILLE?

d. Could you put the work between lines 20 to 24 directly in the kernel?
Write the according code.

e. Write a better CUDA program giving the same result.

(2pts)

(1pt)

(2pts)