Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Dév. GPGPU

Examen — novembre 2019

Durée : 1h30 — Documents autorisés

1 – 10pts Consider the following procedure :

❶ we get two arrays A and B of dimensions 1024x2048 ;

❷ each array contains temperature measures obtained during an experiment in plasma fluid theory ;

❸ we devide the array A into sub-matrices of 8x8, located by k, l in A :

$x_{0,0}$	$x_{0,1}$	$x_{0,2}$	$x_{0,3}$	$x_{0,4}$	$x_{0,5}$	$x_{0,6}$	$x_{0,7}$
$x_{1,0}$	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$	$x_{1,7}$
$x_{2,0}$	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$	$x_{2,7}$
$x_{3,0}$	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$x_{3,6}$	$x_{3,7}$
$x_{4,0}$	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$	$x_{4,5}$	$x_{4,6}$	$x_{4,7}$
$x_{5,0}$	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$	$x_{5,5}$	$x_{5,6}$	$x_{5,7}$
$x_{6,0}$	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$	$x_{6,5}$	$x_{6,6}$	$x_{6,7}$
$x_{7,0}$	$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$x_{7,4}$	$x_{7,5}$	$x_{7,6}$	$x_{7,7}$

where :
- k varies from 0 to 127 ;
- l varies de 0 to 255 ;

❹ for each sub-matrix of A, we compute the arithmetic mean $M_{k,l} = \frac{\sum_{a=0}^{7}\sum_{b=0}^{7} x_{a,b}}{64}$;

❼ we fill the array C of dimensions 1024x2048 according to :
- each value $z_{i,j}$ of C is equal to :

$$z_{i,j} = y_{i,j} - (x_{i,j} - M_{k,l})$$

value in C — $z_{i,j}$; *value in A* — $x_{i,j}$; *value in B* — $y_{i,j}$; *average computed over the corresponding sub-matrix of A* — $M_{k,l}$

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 result** ? *(2pts)*
You will write the new code to add at your program.

2 – 10pts Consider the following program :

```
 1 #include <stdio.h>
 2 #include <cuda.h>
 3
 4 /* removed code */
 5 __global__ void my_kernel(float *t, int r)
 6 {
 7   int position = threadIdx.x + r;
 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;
17   int t = SIZE;
18 /* removed code */
19   cudaMemcpy(gpu_data,data,2*SIZE*sizeof(float),cudaMemcpyHostToDevice);
20   while(t>1)
21   {
22     my_kernel<<<1,SIZE>>>(gpu_data,t);
23     t=t/2;
24   }
25 /* removed code */
26 }
```

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

Questions :

a. Describe and explain **what the program do**. *(2pts)*

b. Are the **memory accesses** performed optimally ? *(1pt)*
Explain your answer.

c. Is the program **easily scalable** : could we expand `TAILLE` ? *(1pt)*

d. Could you put the work between lines 20 to 24 **directly in the kernel** ? *(2pts)*
Write the according code.

e. Write a **better CUDA program** giving the same result. *(4pts)*