Sorbonne Université (Paris 6)
Ingénierie mathématique : Introduction CUDA

Exam 2019

Documents and electronic devices prohibited

Multiple-choice questions (10 points)

Correct answer +1 point, no answer 0 point and wrong answer −0.5 point.

1. What is the maximum value taken by `gridDim.x*threadIdx.x` when we execute `kernel<<<16, 128>>>();` ?
   a) 2048
   b) 2032
   c) 1905

2. The GPU favors an implementation with
   a) very reduced communication between threads
   b) strong communication between threads
   c) either a) or b) according to the algorithm

3. Suppose a block of 64 threads executing the line
   `printf("%i, ", threadIdx.x);`
   We obtain
   a) necessarily : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
   b) very random display, for example : 0, 1, 33, 54, ...
   c) neither options

4. Which of the following syntaxes transfers the 64-bit value from the CPU to the GPU ?
   a) `cudaMemcpy(A, a, 2*sizeof(float), cudaMemcpyHostToDevice);`
   b) `cudaMemcpy(A, a, 2*sizeof(int), cudaMemcpyDeviceToHost);`
   c) `cudaMemcpy(A, a, sizeof(float), cudaMemcpyHostToDevice);`

5. `__syncthreads();` allows to synchronize
   a) each warp
   b) threads
   c) blocks

6. For certain tasks, the GPU is more efficient than the CPU because
   a) it is a coprocessor
   b) its memory has greater bandwidth
   c) its processing units are faster

7. Identify false information
   a) We can launch more blocks than threads per block
   b) The number of threads per block must necessarily be equal to a power of 2
   c) The number of threads per block should preferably be equal to a power of 2

8. `atomicAdd`is a function that

a) allows to make serial an addition
b) allows to use the shared memory
c) allows to make parallel an addition

9. What is the maximum value taken by `blockDim.x*blockIdx.x` when we execute `kernel<<<16, 128>>>();` ?
   a) 2048
   b) 1905
   c) 1920

10. The speedup in Amdahl's law as a function of the number of processors P is
   a) linear
   b) More important than Gustafson's law when $P = 2$
   c) Asymptotically dominated

Exercise : The greatest value of an array (10 points)

Executing `biggest_k<<<numBlocks, threadsPerBlock>>>(int *In, int *Out, int N)`, we want to find the largest element `Out` of an array `In` of `N` unsorted positive integers. To further simplify the exercise, shared memory must have static allocation, for example :
`__shared__ int InSh[64];`
The student can use the same definition of kernel for several questions provided they explain how the definition is generic and has no negative impact on the speed of the calculation.

1. Define the kernel `biggest_k` that uses the maximum number of threads launched in order to find the maximum of the array `In` when executing
   a) `biggest_k<<<1, 1>>>(In, Out, 32);` (1 point)
   b) `biggest_k<<<1, 2>>>(In, Out, 32);` (1 point)
   c) `biggest_k<<<1, 4>>>(In, Out, 32);` (1,5 point)
   d) `biggest_k<<<1, 64>>>(In, Out, 128);` (2 points)

2. Unlike the kernel definition associated with question 1.d, why the one associated with `biggest_k<<<1, 32>>>(In, Out, 64);` or with one of questions 1.a, 1.b, 1.c does not necessarily use `__syncthreads();` ? (1 point)

3. Based on question 1.d, define `biggest_k` to find the maximum of the array `In` when executing `biggest_k<<<1, 64>>>(In, Out, 121);` (1 point)

4. `atomicMax(int* address, int val);` is an atomic function which allows you to do the maximum between `val` and `address[0]` then returns the result in `address`. Based on question 1.d, define `biggest_k` to find the maximum of the array `In` when executing
   a) `biggest_k<<<2, 64>>>(In, Out, 256);` (1,5 points)
   b) `biggest_k<<<(N+127)/128, 128>>>(In, Out, N);` avec `N`> 256 (1 point)