Université de Pau et des Pays de l'Adour
Département de Mathématiques
2022-2023

M2-BigData : GPGPU

Chapter 3 – Exercice 1

Objectives

Lean a basic vector addition with basic thread block and grid dimensions specifications.

Instructions

We are interested on computing

$$C = A + B$$

where A, B and C are vectors of a given size n. To check the correctness of the program, A and B are initialized as follows : $A = (i)_i$ and $B = (n - i)_i$.

Complete the given code `1_vectorAdd.cu` to perform the following algorithm :

- Allocate data on host
- Initialize data on host
- Allocate data on device
- Copy data from host to device
- Compute thread block and kernel grid dimensions
- Invoke the CUDA kernel
- Copy results from device to host
- Verify the result correctness
- Free device memory

To understand the thread block and kernel grid dimensions, you will produce 3 different versions of the program :

1. Use only one block of threads
2. Use only one thread per block
3. Use several threads per block and several blocks

Make sure that your programm is correct for any vector size witout re-compiling.

Questions

1. How many floating operations are being performed in your vector add kernel ? EXPLAIN.
2. How many global memory reads are being performed by your kernel ? EXPLAIN.
3. How many global memory writes are being performed by your kernel ? EXPLAIN.
4. Which version is the most efficient (use a size of $n = 1000$) ? Explain why. Use the NVIDIA profiler to get kernel execution time :
   `nsys profile --stats=true -t cuda ./1-vectorAdd 1000`.

A detailed view of the trace is available using
`nsys stats --report=gputrace report1.qdrep` (Mind the report file name !!)