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

M2-BigData : GPGPU
Chapter 12 – Exercice 1

Objectives

The purpose of this lab is to get you familiar with using the CUDA streaming API by re-implementing a the vector addition lab to use CUDA streams.

Instructions

From the `1-vectorAdd` code (chapter 3) adapt the code with the following modifications :
- Replace host `calloc` by pinned memory host allocation.
- Create arrays of pointer for device memory allocations and an array of `cudaStream_t` for streams. Use `STREAM_NB=4` streams to begin.
- Create the streams using `cudaStreamCreate` function
- Allocate device memory for each buffer in each stream
- Split the computations in a loop over `STREAM_NB*STREAM_SIZE` elements. Each sequence of TransferA, TransferB, add kernel and TransferC is processing `STREAM_NB*STREAM_SIZE` elements.

Indication : you should use two variable for the starting index and the length of the current bloc of elements.

Indication : Mind the total length of the vector : all the memory accesses must be in arrays bounds, use `cuda-memcheck` to be sure.

Questions

1. What is the identifier of the default stream when profiling the initial version of `vectorAdd` from previous lab ?
2. Compare the profiling informations from the Chapter 3 code and your current code, using profiler (`nsys profile` then `nsys-ui`) :
   - What is the speedup of Host-Device transfer speed when using pinned memory .
   - Measure the entire execution time between start of the first copy to device and the end of the last copy from device.
3. Perform a performance study regarding the number of elements computed by each stream (`STREAM_SIZE`). Explain your results and propose a guideline for your next programs using CUDA Streams.