

Assignment I: GPU programming environment

New Attempt

Due 6 Nov 2022 by 23:59 **Points** 1 **Submitting** a file upload **File types** pdf

To submit your assignment, prepare and upload a PDF file fulfilling the following requirements:

- Named as *DD2360HT22_HW1_Surname_Name.pdf*
- You only need to answer **one of the two exercises**
- No code submission is needed for this assignment

For this assignment, you will need to review **Tutorial - [Using CUDA in the laboratory workstations](https://canvas.kth.se/courses/36161/pages/tutorial-using-cuda-in-the-laboratory-workstations)** (<https://canvas.kth.se/courses/36161/pages/tutorial-using-cuda-in-the-laboratory-workstations>) and **Tutorial - [Using CUDA in Google Colab](https://canvas.kth.se/courses/36161/pages/tutorial-using-cuda-in-google-colab)** (<https://canvas.kth.se/courses/36161/pages/tutorial-using-cuda-in-google-colab>).

Exercise 1 - Bandwidth Test GPU-CPU on KTH lab computers

We have learned that one of the current weaknesses of GPU programming is the link between the host and device (GPU) memories. Measure the bandwidth between host-to-device, device-to-host, and device-to-device on Nvidia GPU using the [bandwidthTest](https://github.com/NVIDIA/cuda-samples/tree/master/Samples/1_Uutilities/bandwidthTest) (https://github.com/NVIDIA/cuda-samples/tree/master/Samples/1_Uutilities/bandwidthTest) utility that is included in CUDA SDK. (If you use your own computer, you need to locate it in your own CUDA SDK folder) Follow the instructions to run the bandwidth test and answer questions that are at the end of the report.

1.a - Instructions for measuring bandwidth on KTH lab computers

Setting up the environment

To use the **nvcc** compiler on lab computers, you will need to add the "bin" folder of CUDA to your PATH environment variable. From that point, you can directly use it like any other command:

```
$ export PATH=/usr/local/cuda/bin:$PATH
```

The **-arch** flag is necessary to specify the CUDA architecture that we would like to use. GTX 745 in lab computers use the GM107 processor that belongs to the first generation of Maxwell architecture. It supports compute capability 5.0 and for this reason, we specify **-arch=sm_50**.

Detailed information can be found at [Tutorial - Using CUDA in the laboratory workstations](https://canvas.kth.se/courses/36161/pages/tutorial-using-cuda-in-the-laboratory-workstations) (<https://canvas.kth.se/courses/36161/pages/tutorial-using-cuda-in-the-laboratory-workstations>).

Getting and Running Stream

Copy the [bandwidthTest](http://docs.nvidia.com/cuda/cuda-samples/index.html#bandwidth-test) (<http://docs.nvidia.com/cuda/cuda-samples/index.html#bandwidth-test>) utility from the CUDA SDK examples and compile it:

```
$ cp -rf /usr/local/cuda/samples/1_Uutilities/bandwidthTest ./bandwidthTest
$ cd bandwidthTest
$ nvcc -arch=sm_50 -I/usr/local/cuda/samples/common/inc bandwidthTest.cu -o bandwidthTest
```

The bandwidthTest tool can be executed directly.

```
$ ./bandwidthTest
```

Studying the bandwidth and answering questions

When you execute the program without any arguments you should be able to see something like this:

```
[CUDA Bandwidth Test] - Starting...
Running on...

Device 0: GeForce GTX 745
Quick Mode

Host to Device Bandwidth, 1 Device(s)
PINNED Memory Transfers
  Transfer Size (Bytes)      Bandwidth(GB/s)
  32000000                  13.1

Device to Host Bandwidth, 1 Device(s)
PINNED Memory Transfers
  Transfer Size (Bytes)      Bandwidth(GB/s)
  32000000                  12.7

Device to Device Bandwidth, 1 Device(s)
PINNED Memory Transfers
  Transfer Size (Bytes)      Bandwidth(GB/s)
  32000000                  25.3

Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.
```

The bandwidth test gives you three results: Host to Device, Device to Host and Device to Device. The memory transfer is called Pinned Memory Transfer. We will simply use that and discuss more pinned memory in the latter part of the course (Advanced CUDA). To test for other transfer sizes, you can run the tool in "shmoo" mode:

```
$ ./bandwidthTest --mode=shmoo
```

And you will get something like this for the three kinds of transfer:

```
[CUDA Bandwidth Test] - Starting...
Running on...

Device 0: GeForce GTX 745
Shmoo Mode
```

```

.....
Host to Device Bandwidth, 1 Device(s)
PINNED Memory Transfers
  Transfer Size (Bytes)      Bandwidth(GB/s)
  1000                      0.7
  2000                      1.3
  3000                      2.0
  ...

```

Looking at the results, explain in the report your observations, and why the bandwidth is behaving like that. You can optionally provide a line plot to help your explanation.

Exercise 2 - Bandwidth Test GPU-CPU on Google Colab

To get familiar with the Google Colab environment, find details in Tutorial - [Using CUDA in Google Colab](https://canvas.kth.se/courses/36161/pages/tutorial-using-cuda-in-google-colab) (<https://canvas.kth.se/courses/36161/pages/tutorial-using-cuda-in-google-colab>).

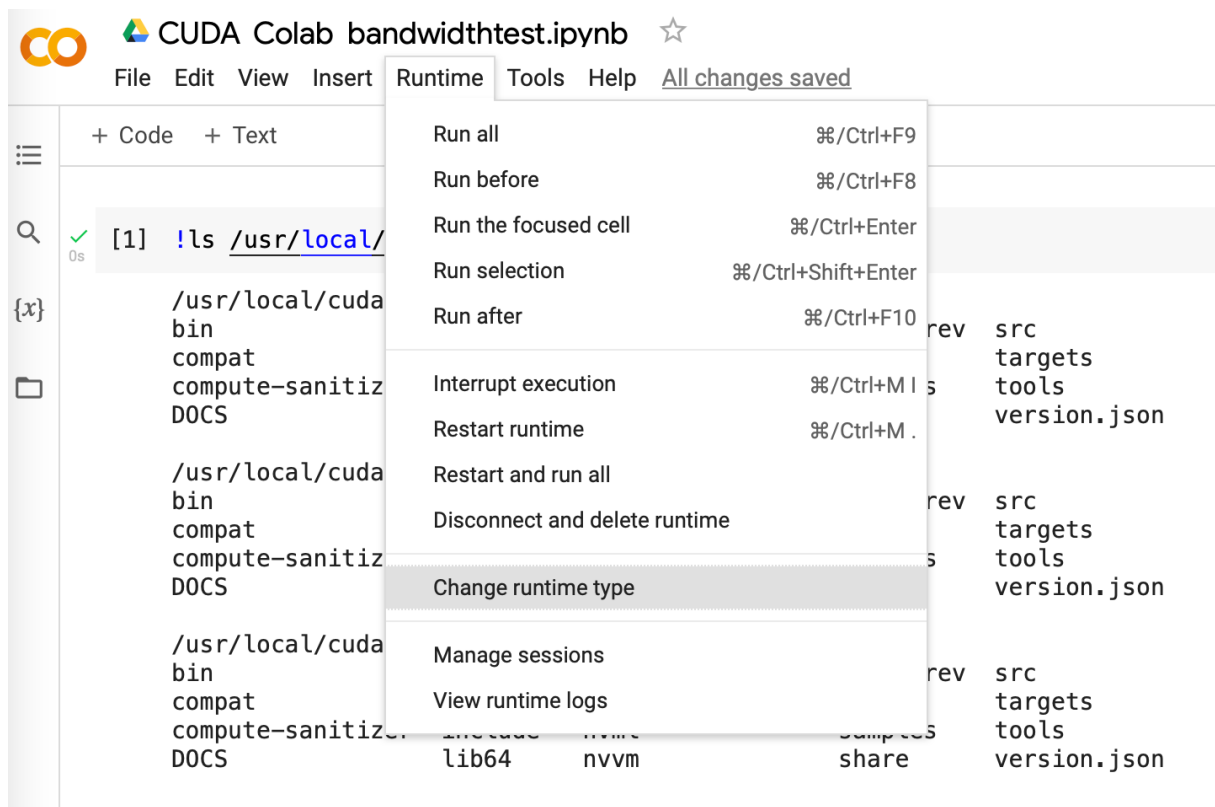
You can refer to this sample notebook to go through the setup:

[CUDA Colab bandwidthtest.ipynb](https://canvas.kth.se/courses/36161/files/5920573?wrap=1) (<https://canvas.kth.se/courses/36161/files/5920573?wrap=1>)

[↓](https://canvas.kth.se/courses/36161/files/5920573/download?download_frd=1) (https://canvas.kth.se/courses/36161/files/5920573/download?download_frd=1)

Setting up the environment

Important note: you need to change your runtime to use GPU as follows:



The screenshot shows the Google Colab interface for a notebook titled 'CUDA Colab bandwidthtest.ipynb'. The 'Runtime' menu is open, displaying various options. The option 'Change runtime type' is highlighted. The background shows a code cell with the command `!ls /usr/local/cuda` and its output, which lists directories like `bin`, `compat`, `compute-sanitizer`, and `DOCS`. The file explorer on the left shows a folder structure with `src`, `targets`, `tools`, and `version.json`.

Notebook settings

None or ?
☒ GPU
TPU

Want access to premium GPUs?
[Purchase additional compute units here.](#)

☐ Omit code cell output when saving this notebook

Cancel Save

Getting and Running Stream

to find out where CUDA SDK examples are located, run this:

CUDA Colab bandwidthtest.ipynb

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

[1]

!ls /usr/local/cuda*

0s

/usr/local/cuda:

bin

compat

compute-sanitizer

DOCS

EULA.txt

extras

include

lib64

libnvvp

nsightee_plugins

nvml

nvvm

nvvm-prev

README

samples

share

src

targets

tools

version.json

/usr/local/cuda-11:

bin

compat

compute-sanitizer

DOCS

EULA.txt

extras

include

lib64

libnvvp

nsightee_plugins

nvml

nvvm

nvvm-prev

README

samples

share

src

targets

tools

version.json

/usr/local/cuda-11.2:

bin

compat

compute-sanitizer

DOCS

EULA.txt

extras

include

lib64

libnvvp

nsightee_plugins

nvml

nvvm

nvvm-prev

README

samples

share

src

targets

tools

version.json

To compile the bandwidth test, you need to copy "bandwidthTest.cu" file to your local directory and then use **nvcc** (do not use the Makefile that is provided inside the folder!):

```
[ ] !cp -rf /usr/local/cuda-11/samples/1_Uutilities/bandwidthTest ./bandwidthTest

[ ] !ls bandwidthTest
!nvcc -I/usr/local/cuda-11/samples/common/inc bandwidthTest/bandwidthTest.cu -o bandwidthTest/bandwidthTest

bandwidthTest.cu Makefile NsightEclipse.xml readme.txt
```

The last step is to execute the bandwidth test:

+ Code + Text

```
▶ !./bandwidthTest/bandwidthTest

[CUDA Bandwidth Test] - Starting...
Running on...

Device 0: Tesla T4
Quick Mode

Host to Device Bandwidth, 1 Device(s)
PINNED Memory Transfers
Transfer Size (Bytes)      Bandwidth(GB/s)
32000000                  12.3

Device to Host Bandwidth, 1 Device(s)
PINNED Memory Transfers
Transfer Size (Bytes)      Bandwidth(GB/s)
32000000                  13.1

Device to Device Bandwidth, 1 Device(s)
PINNED Memory Transfers
Transfer Size (Bytes)      Bandwidth(GB/s)
32000000                  239.3

Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.
```

The bandwidth test gives you three results: Host to Device, Device to Host, and Device to Device. The memory transfer is called Pinned Memory Transfer. We will simply use that and discuss more pinned memory in the [latter part of the course \(Advanced CUDA\)](#). To test for other transfer sizes, you can run the tool in "shmoo" mode. And you will get something like this for the three kinds of transfer:

```
[CUDA Bandwidth Test] - Starting...
Running on...

Device 0: Quadro K420
Shmoo Mode

.....
Host to Device Bandwidth, 1 Device(s)
PINNED Memory Transfers
Transfer Size (Bytes)      Bandwidth(MB/s)
1024                      380.2
2048                      750.8
3072                      1144.0
....
```

Looking at the results, explain in the report your observations, and why the bandwidth is behaving like that. You can optionally provide a line plot to help your explanation.

