HPC - Labwork 2

Tung Nguyen Viet - 2440049

September 2025

1 Introduction

1.1 Device name

```
device = numba.cuda.get_current_device()

print("Device name:", device.name)
print("System specification:", numba.cuda.detect())
```

```
Device name: b'Tesla T4'
Found 1 CUDA devices
id 0             b'Tesla T4'                               [SUPPORTED]
                      Compute Capability: 7.5
                           PCI Device ID: 4
                              PCI Bus ID: 0
                                    UUID: GPU-79dc3cf7-f0f7-f794-86e5-3ed723b9f738
                                Watchdog: Disabled
             FP32/FP64 Performance Ratio: 32
Summary:
        1/1 devices are supported
System specification: True
```

Figure 1: System specification

1.2 Core info: multiprocessor count, core count

```
device = numba.cuda.get_current_device()
print("\n\nMultiprocessors:", device.MULTIPROCESSOR_COUNT)
print("Max threads per block:", device.MAX_THREADS_PER_BLOCK)
my_sms = getattr(device, 'MULTIPROCESSOR_COUNT')
my_cc = device.compute_capability
cc_cores_per_SM_dict = {
    (2,0) : 32,
    (2,1) : 48,
    (3,0) : 192,
    (3,5) : 192,
    (3,7) : 192,
    (5,0) : 128,
    (5,2) : 128,
    (6,0) : 64,
    (6,1) : 128,
    (7,0) : 64,
    (7,5) : 64,
    (8,0) : 64,
    (8,6) : 128,
    (8,9) : 128,
    (9,0) : 128,
    (10,0) : 128,
    (12,0) : 128
    }
cores_per_sm = cc_cores_per_SM_dict.get(my_cc)
total_cores = cores_per_sm*my_sms
print("GPU compute capability: " , my_cc)
print("GPU total number of SMs: " , my_sms)
print("total cores: " , total_cores)
```

```
Multiprocessors: 40
Max threads per block: 1024
GPU compute capability:  (7, 5)
GPU total number of SMs:  40
total cores:  2560
```

Figure 2: Multiprocessor count and core count

Here, I found out this StackOverflow answer, which contains a mapping table to calculate the number of cores.

1.3 Memory info

```python
import numba.cuda

def check_cuda_memory():
    context = numba.cuda.current_context()
    free_mem, total_mem = context.get_memory_info()
    used_mem = total_mem - free_mem


    total_mem_gb = total_mem / (1024**3)
    free_mem_gb = free_mem / (1024**3)
    used_mem_gb = used_mem / (1024**3)

    print(f"CUDA Device: {context.device.name}")
    print(f"Total CUDA Memory: {total_mem_gb:.2f} GB")
    print(f"Free CUDA Memory: {free_mem_gb:.2f} GB")
    print(f"Used CUDA Memory: {used_mem_gb:.2f} GB")

if __name__ == "__main__":
    check_cuda_memory()
```

```
CUDA Device: b'Tesla T4'
Total CUDA Memory: 14.74 GB
Free CUDA Memory: 14.64 GB
Used CUDA Memory: 0.10 GB
```

Figure 3: GPU memory info