

# Lab05: Numba

---

Felix <chengzongli.ee10@nycu.edu.tw>

# Outline

- GPU Introduction
  - What is GPU?
  - GPU vs CPU
  - GPU On Your Computer
  - NVIDIA GPU Architecture
- What is CUDA?
- Heterogeneous Programming
- Tools to check GPU(s) and the % of utilization
- CMake
- Example
- Exercise

# Example: Use numba to invoke the kernel function

## Writing a kernel function using cuda

- Uses **@cuda.jit** decorator to indicate that it's a kernel function
- **cuda.grid(N)** will calculate the index in **N-dim** space for you
  - Use **cuda.threadIdx.x**, **cuda.blockIdx.x** is also OK

```
from numba import cuda

@cuda.jit
def add_kernel(x, y, out):
    block_size = cuda.blockDim.x
    grid_size = cuda.gridDim.x

    start = cuda.grid(1)
    stride = block_size * grid_size
    for i in range(start, x.shape[0], stride):
        out[i] = x[i] + y[i]
```

# Example: Use numba to invoke the kernel function

## Invoking kernel function

- Allocate memory on device
- Move data to device

```
import numpy as np

n = 100000
x = np.arange(n).astype(np.float32) / 1000.0
y = 2 * x

d_x = cuda.to_device(x)
d_y = cuda.to_device(y)
d_out = cuda.device_array(n, dtype=np.float32)
```

# Example: Use numba to invoke the kernel function

## Invoking kernel function

- Invoke kernel function
  - block/thread in 2 or 3-D  
uses tuple instead of just an integer
- Synchronize the threads
  - **copy\_to\_host** implicitly does that
  - therefore it is not necessary in this code

```
threads_per_block = 256
blocks_per_grid = 30

add_kernel [blocks_per_grid, threads_per_block] (
    d_x, d_y, d_out
)

cuda.synchronize()
out = d_out.copy_to_host()
print(out[:20])
```

# Example: Use numba to invoke the kernel function

In numba, this code works also

- **without explicitly moving data between host and device**

```
import numpy as np

n = 100000
x = np.arange(n).astype(np.float32) / 1000.0
y = 2 * x
out = np.empty_like(x)

threads_per_block = 128
blocks_per_grid = 30

add_kernel[blocks_per_grid, threads_per_block](x, y, out)
```

# Exercise: Grayscale the image!

Please refer to the instructions in the jupyter notebook file

Original Image



Grayscaled image using CPU



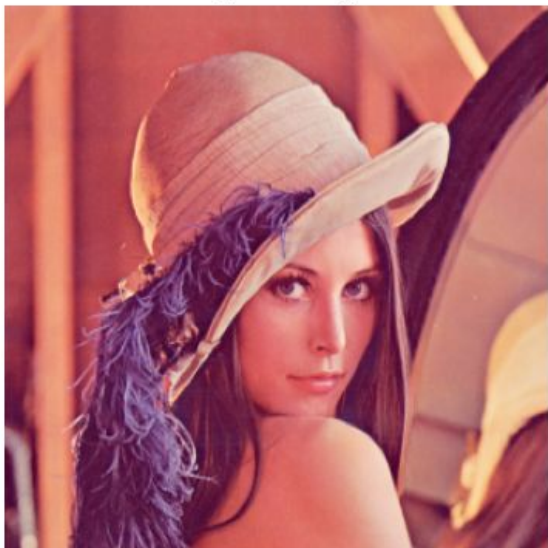
Grayscaled image using GPU



# Exercise: Image Convolution

Please refer to the instructions in the jupyter notebook file

Original Image



Blurred Image

