Advaned Programming fo HPC - Report 3

NGUYEN TAT HUNG

December 24, 2022

Implementation

```
import sys
import numba
from numba import cuda
import numpy as np
from numba import vectorize
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from numpy import asarray

print("Python version:", sys.version)
print("Numba version:", numba.__version__)
print("Numpy version:", np.__version__)
print("Numpy version:",cuda.gpus.__module__)


#img = mpimg.imread('example-orig.jpg')
#imgplot = plt.imshow(img)
#plt.show()

#x_data = asarray(mpimg.imread('example-orig.jpg'))
#flatSrc = x_data.flatten()

img = cv.imread("example-orig.jpg")

im_resized = cv.resize(img, (224, 224), interpolation=cv.INTER_LINEAR)
plt.imshow(cv.cvtColor(im_resized, cv.COLOR_BGR2RGB))
plt.axis('off')
plt.show()

x_data = np.array(img)
flatSrc = x_data.flatten()

imageWidth,imageHeight,c = x_data.shape
print(x_data.shape)

@cuda.jit
def rgb2gray(src, dst):
  tidx = cuda.threadIdx.x + cuda.blockIdx.x * cuda.blockDim.x
  tidy = cuda.threadIdx.y + cuda.blockIdx.y * cuda.blockDim.y
```

```
    g = np.uint8((src[tidx,tidy, 0] + src[tidx,tidy, 1] + src[tidx,tidy, 2]) / 3)
    dst[tidx,tidy, 0] = dst[tidx,tidy, 1] = dst[tidx,tidy, 2] = g

pixelCount = imageWidth * imageHeight
#blockSize = 64
#gridSize = int(pixelCount / blockSize)
gridSize = (8, 8);
blockSize = (32, 32);

devSrc = cuda.to_device(flatSrc)
devDst = cuda.device_array((pixelCount, 3), np.uint8)
rgb2gray[gridSize, blockSize](devSrc, devDst)
hostDst = devDst.copy_to_host()

print(flatSrc.shape)
```

Result

Figure 1: Original input image

Figure 2: Output image