Lab 1

Keith Evan Schubert

June 9, 2025

1 Objective

The purpose of this lab is to get you familiar with using the CUDA API by implementing a simple vector addition kernel and its associated setup code. Note: This is a simple but essential exercise. Please write out the code and do not copy it from other examples or lecture slides. That process is most important.

2 Activity

1. Login to kodiak. cd to your mpplabs directory and type `git pull`.
2. Edit the file `<lab-directory>/main.cu` to implement the following where indicated:
   (a) Allocate device memory
   (b) Copy host memory to device
   (c) Initialize thread block and kernel grid dimensions and invoke CUDA kernel
   (d) Copy results from device to host
   (e) Free device memory
3. Edit the file ¡lab-directory¿/kernel.cu to implement the vector addition kernel code.
4. Compile and test your code. Try it for several sizes, say 1k, 10k, 100k, 1M. How does the time change? Does each part change the same?

```
cd <lab-directory>
make
nano vecadd.sh                 #add the following
  ~/,lab-directory>/vecadd      # Uses the default vector size
  ~/,lab-directory>/vecadd <m>  # Uses vectors of size m
mpprun vecadd.sh
```

3 Turn in

Upload a report that includes the output with analysis of the time complexity scaling, main.cu, and kernel.cu to the course Canvas site.

4 Going Further

Try this same thing for changing an image to grayscale.