# Lab 3

Keith Evan Schubert

June 9, 2025

## 1 Objective

The purpose of this lab is to implement a tiled dense matrix multiplication routine.

## 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) Copy results from device to host
   (d) Free device memory
3. Edit the file `<lab-directory>/kernel.cu` to initialize the thread block and kernel grid dimensions and invoke the CUDA kernel, and to implement the tiled matrix multiplication kernel code..
4. Compile and test your code.

```
cd <lab-directory>
make
nano sgemm-tiled.sh  # add sgemm-tiled commands per below
    ~/<lab-directory>/sgemm-tiled   # Uses the default matrix sizes
    ~/<lab-directory>/sgemm-tiled <m>   # Uses square m x m matrices
    ~/<lab-directory>/sgemm-tiled <m> <k> <n> # Uses (m x k) and
                                  # (k x n) input matrices
mpprun sgemm-tiled.sh
```

# 3 Turn in

Upload to the course Canvas site:

1. a report that includes :
   (a) the output
   (b) analysis of the performance: Try the code for several sizes, square and non-square matrices, and matrices that fit and don't fit (neatly) in the blocks How does the time change? Does each part change the same?
   (c) answer section where you answer the following:
      i. In your kernel implementation, how many threads can be simultaneously scheduled for execution on a GeForce GTX 280 GPU, which contains 30 streaming multiprocessors? Use:

         ```
         nvcc --ptxas-options="-v" kernel.cu
         ```

         to see the resource usage of your kernel (although compilation will fail, it will only do so after compiling the kernel and displaying the relevant information). Show your work.
2. main.cu
3. kernel.cu

The cuda code will be graded for completeness, correctness, handling of boundary, and style (5pts). The report will be graded on readability, clarity, analysis, and solution to the questions (5pts).

# 4 Going Further

There should be a definite improvement in time but the complexity should stay the same (how it grows relatively). Think about at what point matrix multiply in this fashion becomes too long. 1 day? 1 week? Estimate the problem size.