Hassan Ahmad

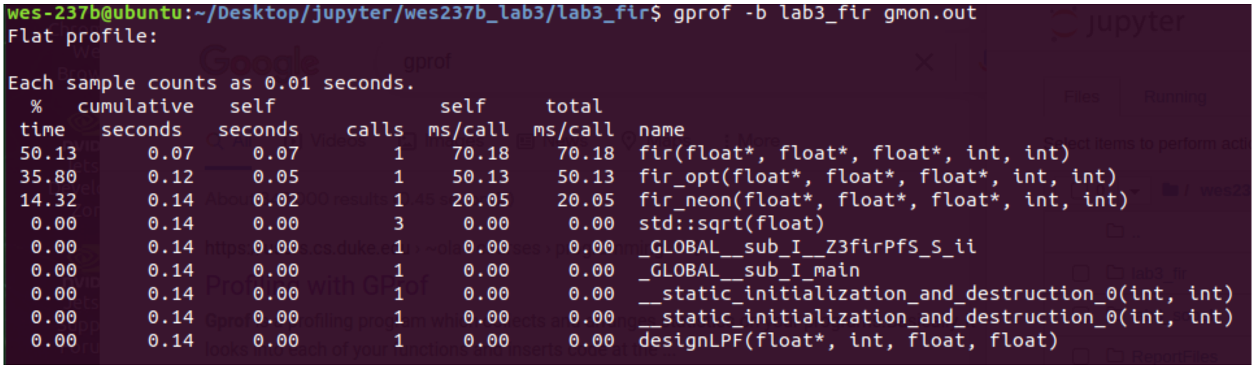
ID: 59014371

WES 237B: Assignment 3 Report

Lab Deliverables:

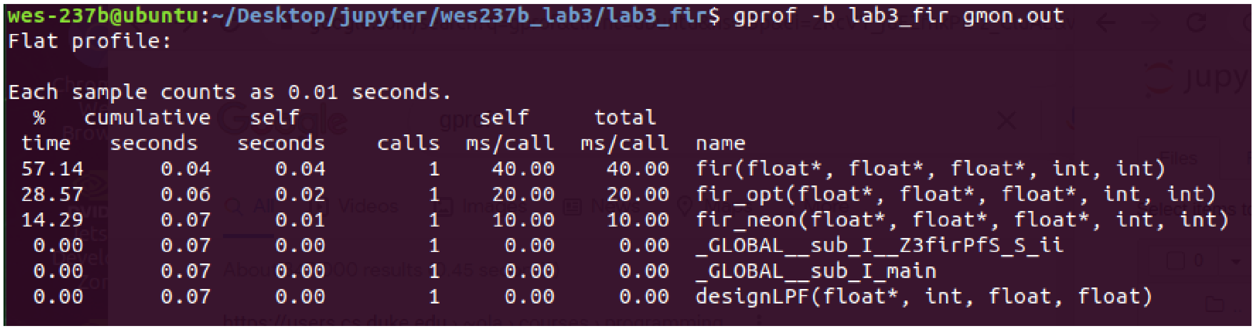
1. Compiling with *-O0*, *-O1*, *-O2 gprof* comparison

   Using *-O0*:



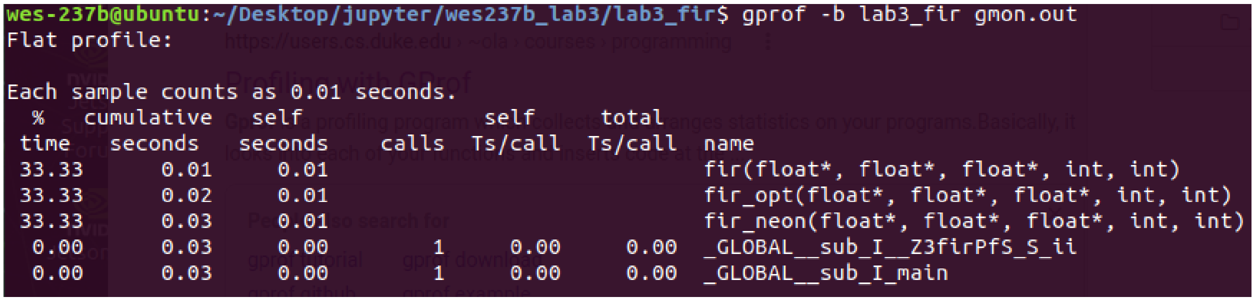
```
wes-237b@ubuntu:~/Desktop/jupyter/wes237b_lab3/lab3_fir$ gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
 50.13      0.07     0.07        1    70.18    70.18  fir(float*, float*, float*, int, int)
 35.80      0.12     0.05        1    50.13    50.13  fir_opt(float*, float*, float*, int, int)
 14.32      0.14     0.02        1    20.05    20.05  fir_neon(float*, float*, float*, int, int)
  0.00      0.14     0.00        3     0.00     0.00  std::sqrt(float)
  0.00      0.14     0.00        1     0.00     0.00  _GLOBAL__sub_I__Z3firPfS_S_ii
  0.00      0.14     0.00        1     0.00     0.00  _GLOBAL__sub_I_main
  0.00      0.14     0.00        1     0.00     0.00  __static_initialization_and_destruction_0(int, int)
  0.00      0.14     0.00        1     0.00     0.00  __static_initialization_and_destruction_0(int, int)
  0.00      0.14     0.00        1     0.00     0.00  designLPF(float*, int, float, float)
```

   Using *-O1*:



```
wes-237b@ubuntu:~/Desktop/jupyter/wes237b_lab3/lab3_fir$ gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
 57.14      0.04     0.04        1    40.00    40.00  fir(float*, float*, float*, int, int)
 28.57      0.06     0.02        1    20.00    20.00  fir_opt(float*, float*, float*, int, int)
 14.29      0.07     0.01        1    10.00    10.00  fir_neon(float*, float*, float*, int, int)
  0.00      0.07     0.00        1     0.00     0.00  _GLOBAL__sub_I__Z3firPfS_S_ii
  0.00      0.07     0.00        1     0.00     0.00  _GLOBAL__sub_I_main
  0.00      0.07     0.00        1     0.00     0.00  designLPF(float*, int, float, float)
```

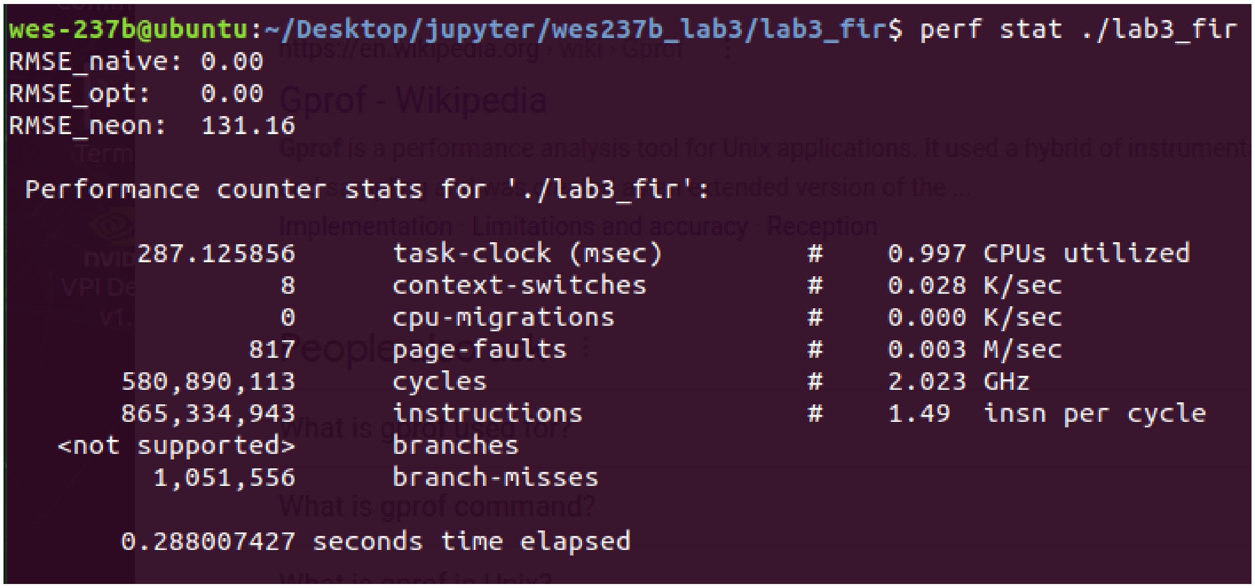
   Using *-O2*:



```
wes-237b@ubuntu:~/Desktop/jupyter/wes237b_lab3/lab3_fir$ gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  Ts/call  Ts/call  name    
 33.33      0.01     0.01                             fir(float*, float*, float*, int, int)
 33.33      0.02     0.01                             fir_opt(float*, float*, float*, int, int)
 33.33      0.03     0.01                             fir_neon(float*, float*, float*, int, int)
  0.00      0.03     0.00        1     0.00     0.00  _GLOBAL__sub_I__Z3firPfS_S_ii
  0.00      0.03     0.00        1     0.00     0.00  _GLOBAL__sub_I_main
```

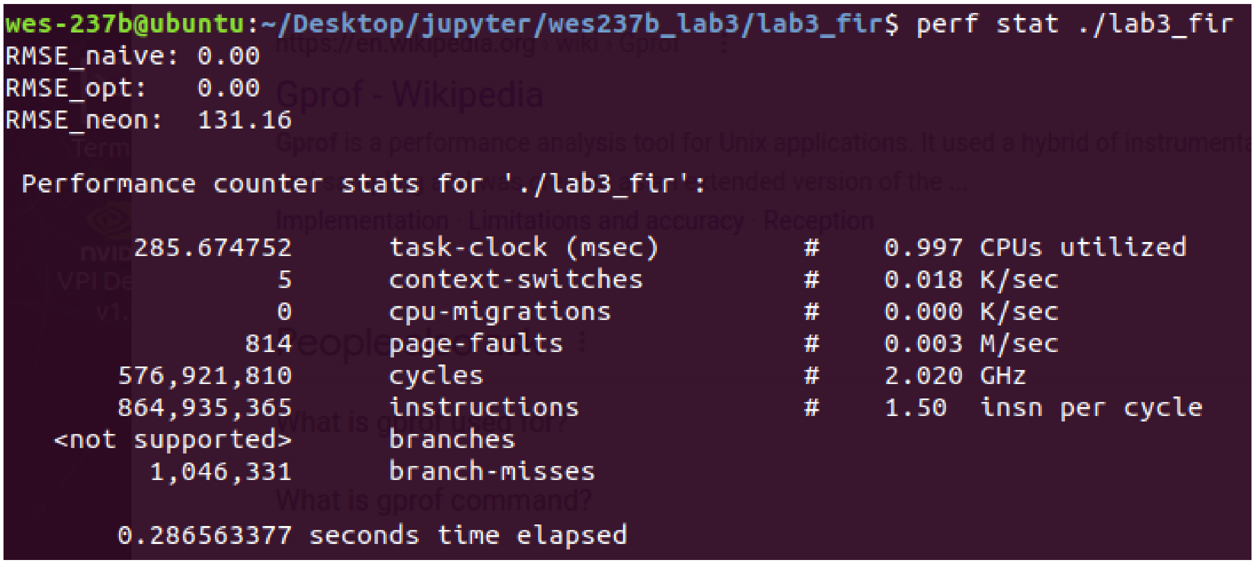
2. Compiling without -pg:

   With *-pg* (including gprof execution):



```
wes-237b@ubuntu:~/Desktop/jupyter/wes237b_lab3/lab3_fir$ perf stat ./lab3_fir
RMSE_naive: 0.00
RMSE_opt:   0.00
RMSE_neon:  131.16

 Performance counter stats for './lab3_fir':

        287.125856      task-clock (msec)         #    0.997 CPUs utilized
                 8      context-switches          #    0.028 K/sec
                 0      cpu-migrations            #    0.000 K/sec
               817      page-faults               #    0.003 M/sec
       580,890,113      cycles                    #    2.023 GHz
       865,334,943      instructions              #    1.49  insn per cycle
   <not supported>      branches
         1,051,556      branch-misses

       0.288007427 seconds time elapsed
```

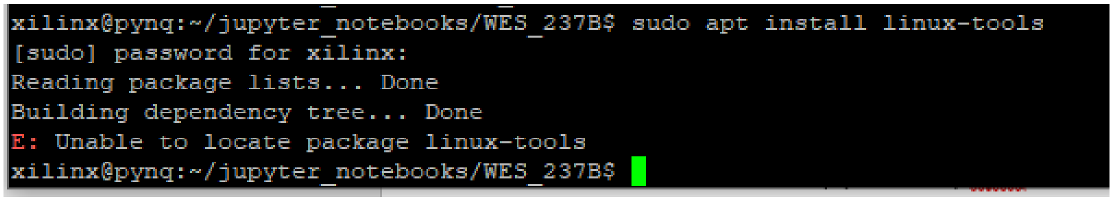
   Without *-pg* (not including gprof execution):



```
wes-237b@ubuntu:~/Desktop/jupyter/wes237b_lab3/lab3_fir$ perf stat ./lab3_fir
RMSE_naive: 0.00
RMSE_opt:   0.00
RMSE_neon:  131.16

 Performance counter stats for './lab3_fir':

        285.674752      task-clock (msec)         #    0.997 CPUs utilized
                 5      context-switches          #    0.018 K/sec
                 0      cpu-migrations            #    0.000 K/sec
               814      page-faults               #    0.003 M/sec
       576,921,810      cycles                    #    2.020 GHz
       864,935,365      instructions              #    1.50  insn per cycle
   <not supported>      branches
         1,046,331      branch-misses

       0.286563377 seconds time elapsed
```

3. Comparing platforms:

   I could not set up perf on my Pynq Z2 board. When I run *sudo apt install linux-tools* to set up Perf, I get this error:



```
xilinx@pynq:~/jupyter_notebooks/WES_237B$ sudo apt install linux-tools
[sudo] password for xilinx:
Reading package lists... Done
Building dependency tree... Done
E: Unable to locate package linux-tools
xilinx@pynq:~/jupyter_notebooks/WES_237B$
```

Assignment Deliverables:

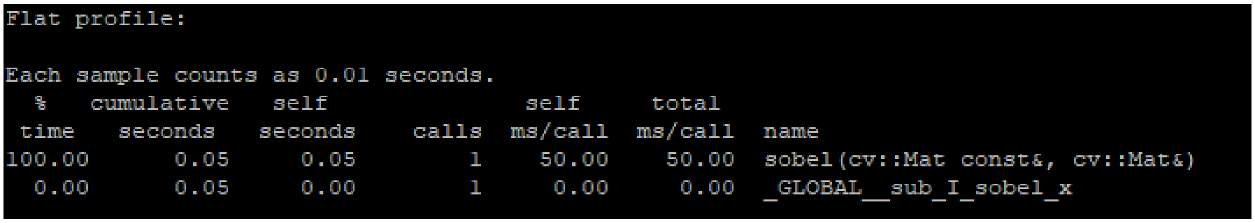
1. NEON Intrinsics for Optimized Sobel

   In my implementation of the optimized Sobel filter, I attempted to vectorize the pixels of the input image that is being indexed across, multiplied, and accumulated in the algorithm. I created uint8x16_t data type variables to accommodate the 8-bit uchar data type of the grayscale image pixels. My vectorization would load 16 separate pixel values and execute the multiplication with a uint8x16_t vectorized variable with all 16 'lanes' containing the exact same value.

   Note: My implementation of the Neon-optimized sobel filter is not complete; I was having difficulty resolving the task of loading each pixel value into a Neon intrinsic datatype, while also accessing the neighboring pixel values to multiply and accumulate as the resultant magnitude. The task made sense conceptually, but I had trouble implementing the above in a valid iterative method.

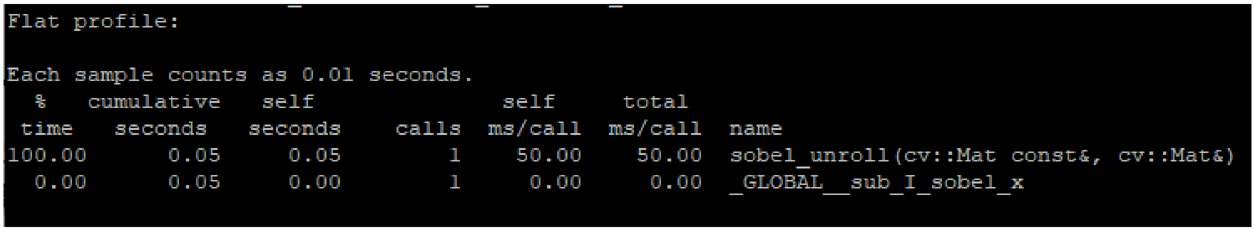
2. Comparison between baseline, loop-unrolled, and NEON Sobel (gprof or perf or etc)
   Using gprof to compare between baseline sobel (naïve) and unrolled sobel:

   Naïve Sobel: 50ms exec. time



```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
100.00      0.05     0.05        1    50.00    50.00  sobel(cv::Mat const&, cv::Mat&)
  0.00      0.05     0.00        1     0.00     0.00  _GLOBAL__sub_I_sobel_x
```

   Unrolled Sobel: 50ms exec. time



```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
100.00      0.05     0.05        1    50.00    50.00  sobel_unroll(cv::Mat const&, cv::Mat&)
  0.00      0.05     0.00        1     0.00     0.00  _GLOBAL__sub_I_sobel_x
```

3. Extra: use gcov to view code coverage

```
xilinx@pynq:~/jupyter_notebooks/WES_237B/lab3_sobel/objs$ sudo gcov main.o

File 'src/main.cpp'

Lines executed:70.45% of 44

Creating 'main.cpp.gcov'

Cannot open source file src/main.cpp


File '/usr/include/c++/9/iostream'

No executable lines

Removing 'iostream.gcov'


File '/usr/include/c++/9/bits/basic_string.tcc'

Lines executed:0.00% of 9

Creating 'basic_string.tcc.gcov'


File '/usr/include/c++/9/bits/stl_iterator_base_funcs.h'

Lines executed:0.00% of 1

Creating 'stl_iterator_base_funcs.h.gcov'


File '/usr/include/c++/9/bits/basic_string.h'

Lines executed:44.44% of 9

Creating 'basic_string.h.gcov'


File '/usr/include/c++/9/bits/char_traits.h'

Lines executed:0.00% of 3

Creating 'char_traits.h.gcov'


File '/usr/include/c++/9/ext/string_conversions.h'
```

```
Lines executed:75.00% of 12
Creating 'string_conversions.h.gcov'

File '/usr/include/c++/9/ext/new_allocator.h'
Lines executed:100.00% of 1
Creating 'new_allocator.h.gcov'

File '/usr/include/stdlib.h'
Lines executed:0.00% of 1
Creating 'stdlib.h.gcov'

File '/usr/include/opencv2/core/mat.inl.hpp'
Lines executed:86.67% of 30
Creating 'mat.inl.hpp.gcov'

File '/usr/include/opencv2/core/types.hpp'
Lines executed:100.00% of 4
Creating 'types.hpp.gcov'

File '/usr/include/c++/9/bits/stl_vector.h'
Lines executed:100.00% of 4
Creating 'stl_vector.h.gcov'

File '/usr/include/c++/9/bits/alloc_traits.h'
Lines executed:0.00% of 1
Creating 'alloc_traits.h.gcov'

File '/usr/include/arm-linux-gnueabihf/bits/stdio2.h'
Lines executed:100.00% of 3
Creating 'stdio2.h.gcov'
```

```
File '/usr/include/opencv2/core/mat.hpp'
No executable lines
Removing 'mat.hpp.gcov'

File '/usr/include/c++/9/ostream'
Lines executed:0.00% of 2
Creating 'ostream.gcov'

File '/usr/include/c++/9/bits/locale_facets.h'
Lines executed:0.00% of 5
Creating 'locale_facets.h.gcov'

xilinx@pynq:~/jupyter_notebooks/WES_237B/lab3_sobel/objs$
```

Using GCOV to highlight code usage in my program, I can see that there are a few header files that do not get used during execution. Additionally, there are some lines in my main.cpp file that are not used. This is likely because the file execution calls only one of the three methods for exercising the sobel filter on the input image, which I select as an argument when running the executable.

Introduction to CUDA:

1. The flag **global** indicates a Cuda function that can be called by the host, which is the CPU in our case. With this flag, we can invoke the GPU to complete tasks from some other execution on the CPU.
2. The code:

   ```
   foo<<4,32>>(out, in1, in2)
   ```

   defines a function 'foo' that has a grid consisting of 4 blocks, where each block contains 32 threads. Each thread is running the same computation on different sets of data. Each block is run on a different multiprocessor in the GPU. The parameters of the function 'foo' are pointer variables to the source and destination arrays. The first parameter, 'out', is a pointer to the destination array, and following two parameters are pointers to source arrays that would likely be needed for some calculation.