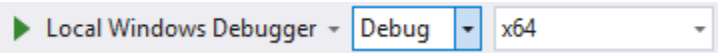
Guide on Debugging, Testing, Submitting and Profiling CUDA Project

This tutorial shows you how to debug, test, profiling your code as well as building your work for submitting on e-class.

This tutorial is applicable for lab 2 to 10. For each lab from 2 to 10 you will be given a Visual Studio 2013 project.

Debugging

To debug your program, first you must choose and build **Debug** configuration:



Change your configuration to Debug in the drop list and run BUILD->Build Solution

Then you can set break points in your program and start debugging.

For normal code (i.e. code that are not kernel or device code, for example the main function), you just have to run DEBUG->Start Debugging like you would normally do in Visual Studio.

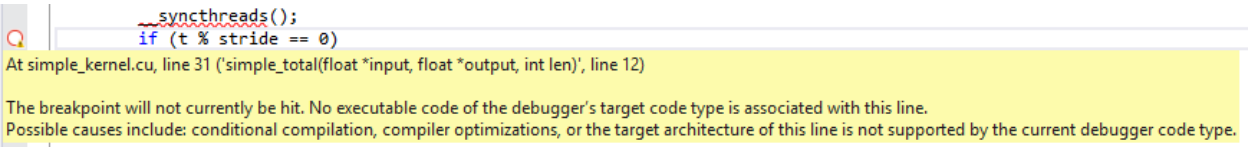
For kernel or device code, you must run the debugger from NSIGHT. For example, let say you have a kernel like this with a break point:

```
__global__ void simple_total(float *input, float *output, int len) {
    // TODO: Load a segment of the input vector into shared memory
    __shared__ float partialSum[2 * BLOCK_SIZE];
    unsigned int t = threadIdx.x;
    unsigned int start = 2 * blockIdx.x*BLOCK_SIZE;
    partialSum[t] = input[start + t];
    partialSum[BLOCK_SIZE + t] = input[start + BLOCK_SIZE + t];

    for (unsigned int stride = 1; stride <= blockDim.x; stride *= 2)
    {
        __syncthreads();
        if (t % stride == 0)
            partialSum[2 * t] += partialSum[2 * t + stride];
    }

    // TODO: Write the computed sum of the block to the output vector at the
    // correct index
    if (t == 0)
        output[blockIdx.x] = partialSum[0];
}
```

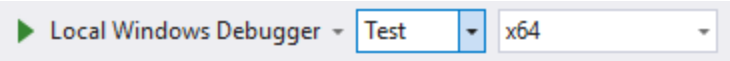
If you use the normal debugger Visual Studio will show this message and will not allow you to debug:



Run the NSIGHT Debugger in NSIGHT->Start CUDA Debugging and you can debug the code like you normally do.

Testing

To test your program, change the configuration to **Test** and build it.



After that the directions on how to continue testing will be given and vary for each lab. Typically you will be given a test dataset and a .bat file that do the test automatically

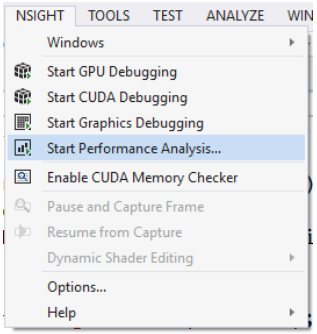
Important: To make sure all testing run correctly the path to your project must NOT contain any space in it.

Submitting your program

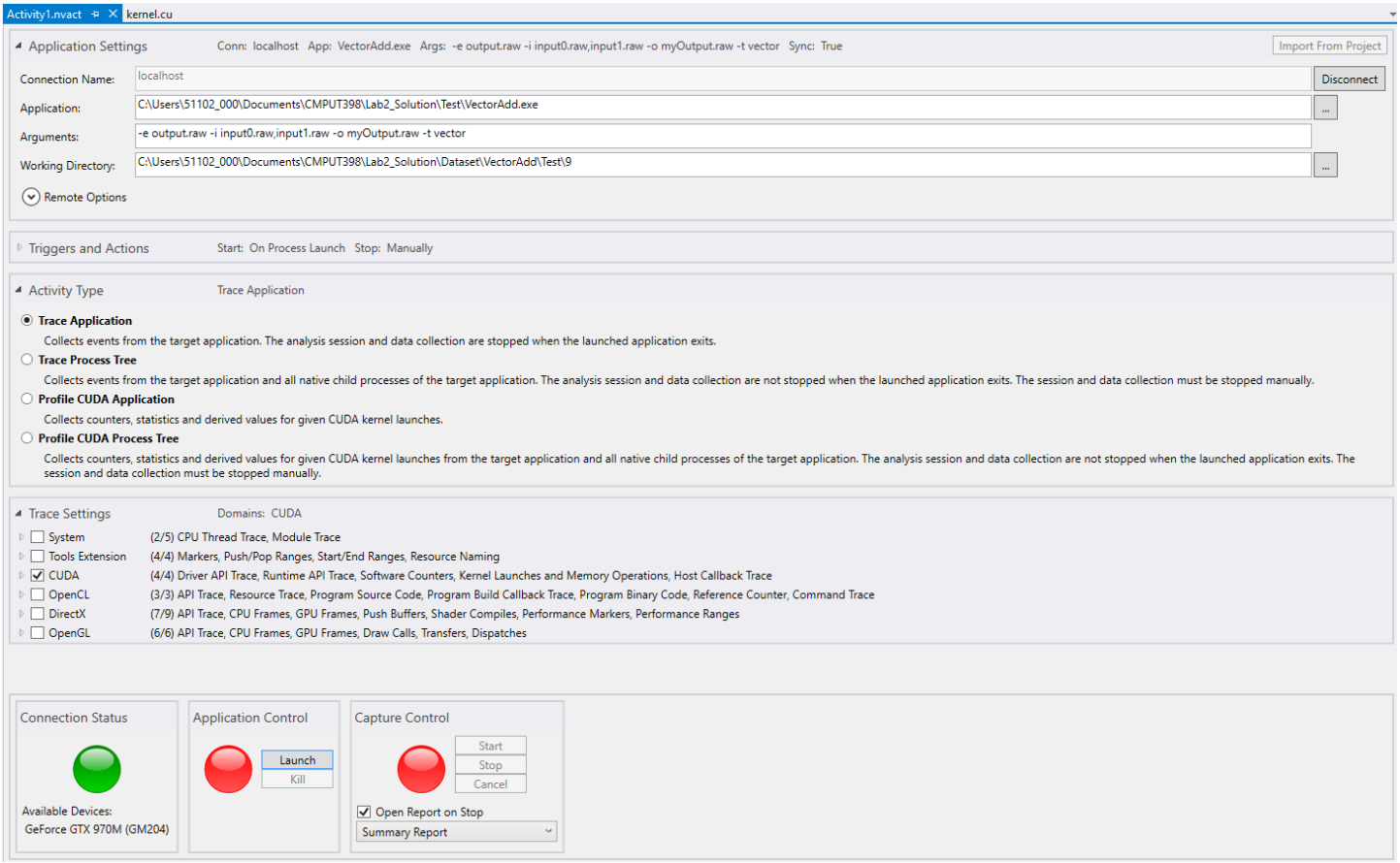
To submit your program, you must build your project using the Submission configuration. A directory with name "Submission" will be created inside your project directory. Zip and submit this folder as well as additional files (if applicable)

Profiling your program using NSIGHT

To profile your program, you must first build the Test configuration. **Click NSIGHT->Start Performance Analysis…**



Then, a windows like this will be opened:



First, in “Application Setting”:

- Enter the executable file that was built by the **Test** configuration. For example in lab 2 it will be in <Path to your project>\Test\VectorAdd.exe
- Enter the correct arguments. This changes for different project.
- Working directory is the directory of the test case. They will be in the directory: <Path to your project>Dataset\Test\<Test case number>

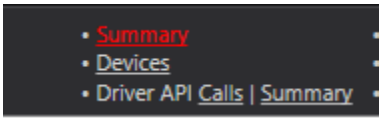
For each project, you will be provided with the instructions how to set the arguments and the working directory correctly.

Then, in “Activity Type”, choose “Trace Application”

In Trace Setting, tick “CUDA”

Finally, click “Launch” button in “Application Control”.

After it finishes running, you will get an overview windows. Under “CUDA Overview” there will be a link to summary:

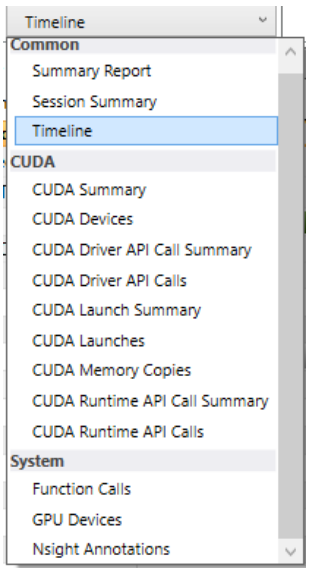


Click on it and it will open a page that shows you many details including the total running time of your program, for example:

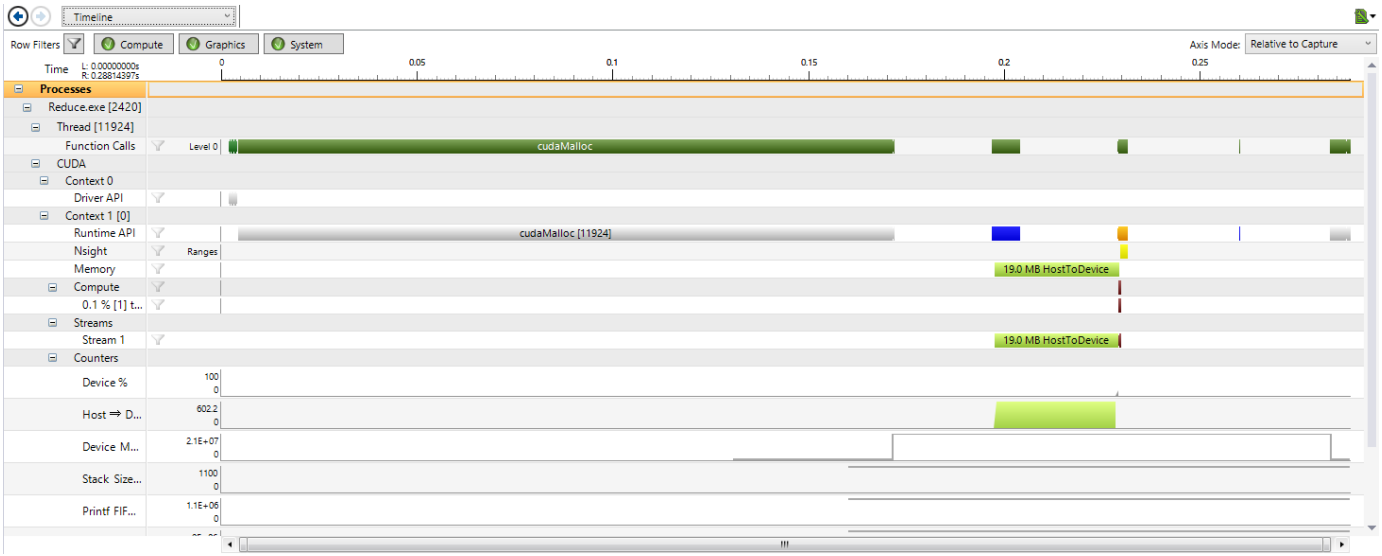
Top Device Functions By Total Time Summary | All

	Name	Launches	Device %	Total (µs)	Min (µs)	Avg (µs)	Max (µs)
1	total	1	0.15	418.259	418.259	418.259	418.259

Or you can choose from the drop-down list “Timeline” to see a graphical summary of your program:



It will show a page similar to this:


Timeline
Row Filters
Compute
Graphics
System
Axis Mode:
Relative to Capture
Time
L: 0.00000000s
R: 0.28814397s
0
0.05
0.1
0.15
0.2
0.25
Processes
Reduce.exe [2420]
Thread [11924]
Function Calls
Level 0
cudaMalloc
CUDA
Context 0
Driver API
Context 1 [0]
Runtime API
cudaMalloc [11924]
Nsight
Ranges
Memory
19.0 MB HostToDevice
Compute
0.1 % [1] t...
Streams
Stream 1
19.0 MB HostToDevice
Counters
Device %
100
0
Host ⇒ D...
602.2
0
Device M...
2.1E+07
0
Stack Size...
1100
0
Printf FIF...
1.1E+06
0