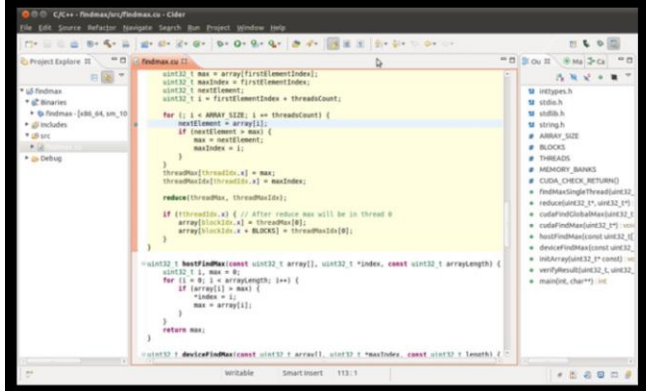
CUDA Profile

ECE 277

Cheolhong An

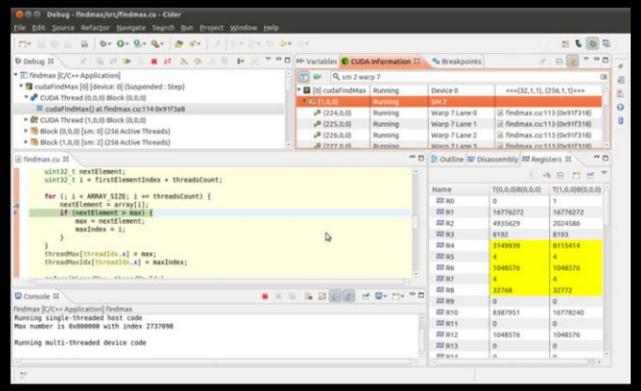
NVIDIA® NSIGHT™ eclipse

Full featured IDE for developing CUDA-C apps on X86, ARM & POWER8.



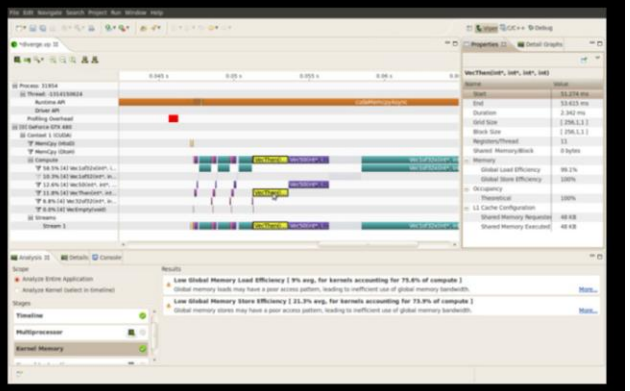
CUDA-Aware Editor

- Automated CPU to GPU code refactoring
- Semantic highlighting of CUDA code
- Integrated code samples & docs
- Cross-compilation for Linux target



Nsight Debugger

- Simultaneously debug CPU and GPU code
- Inspect variables across CUDA threads
- Use breakpoints & single-step debugging
- Integrated CUDA memory checker



Nsight Profiler

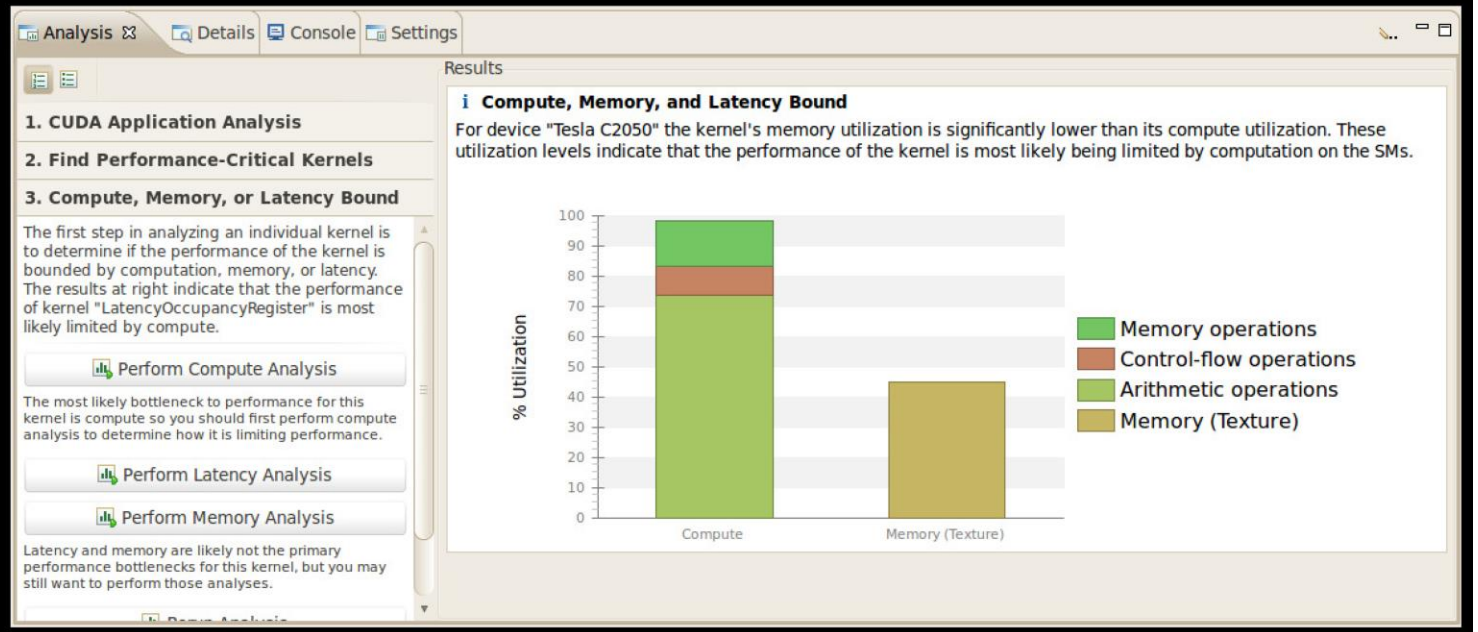
- Quickly identifies performance issues
- Guided expert analysis
- Source line correlation

CUDA STANDALONE TOOLS

Independent (not integrated)

Visual Profiler

- Trace CUDA activities
- Performance instrumentation with source code correlation
- Guided Expert Analysis



NVPROF

- Generates execution summary
- Gather Performance events

CUDA-MEMCHECK

- Out of bounds memory access detection
- Detects Race Condition

CUDA-GDB

- Command line CUDA debugging
- Debug CPU and GPU code

* Android new in CUDA 6.0

NVIDIA® NSIGHT™ VISUAL STUDIO EDITION

Visual Studio integrated development for GPU and CPU

Kernel launch cost ⇒ Heavy 작업임.

- Can you guess how long does it take to launch an empty kernel?

  w/o any task (just invoke)

- Let's check it out

- So Can you guess what size of your problem is good at GPU instead of CPU?

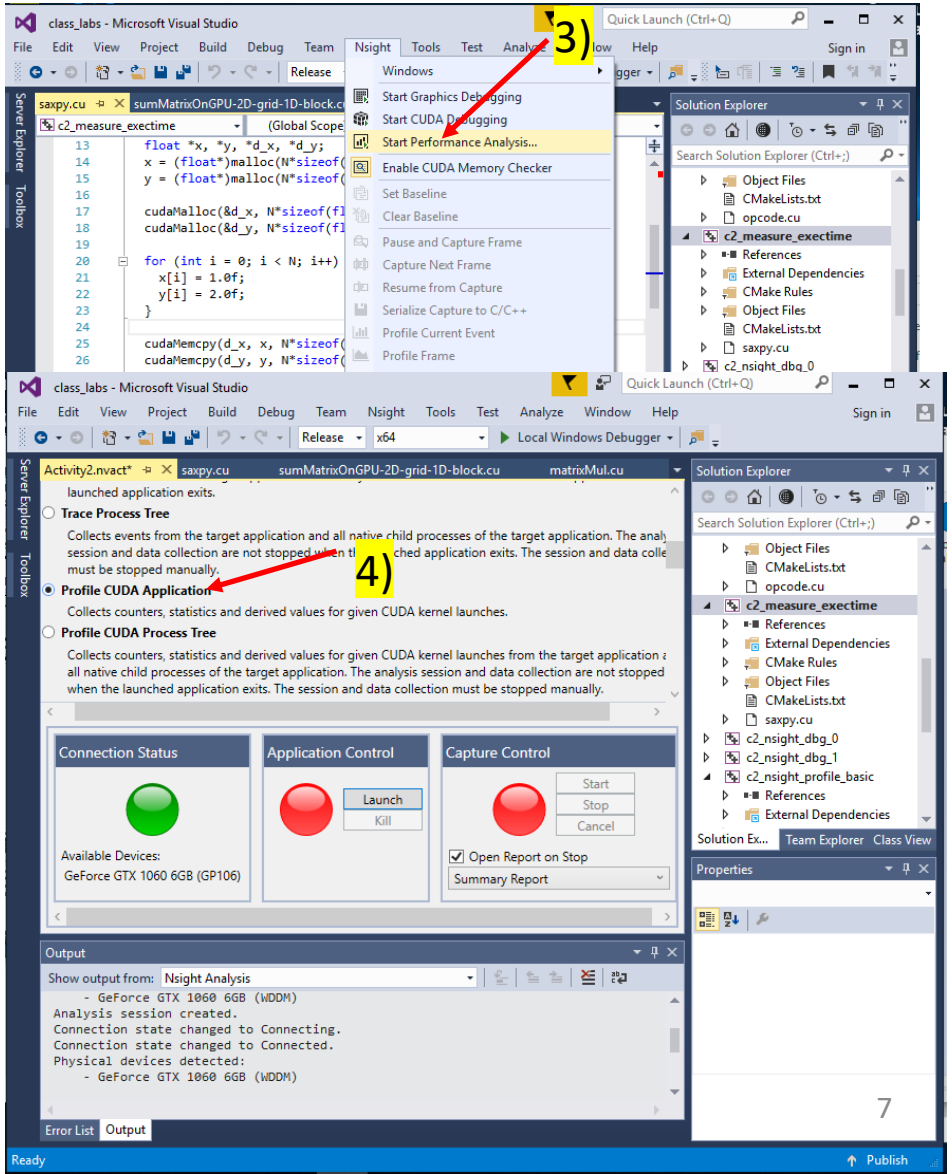
How to measure execution time

1. CUDA profiler

2. Event method

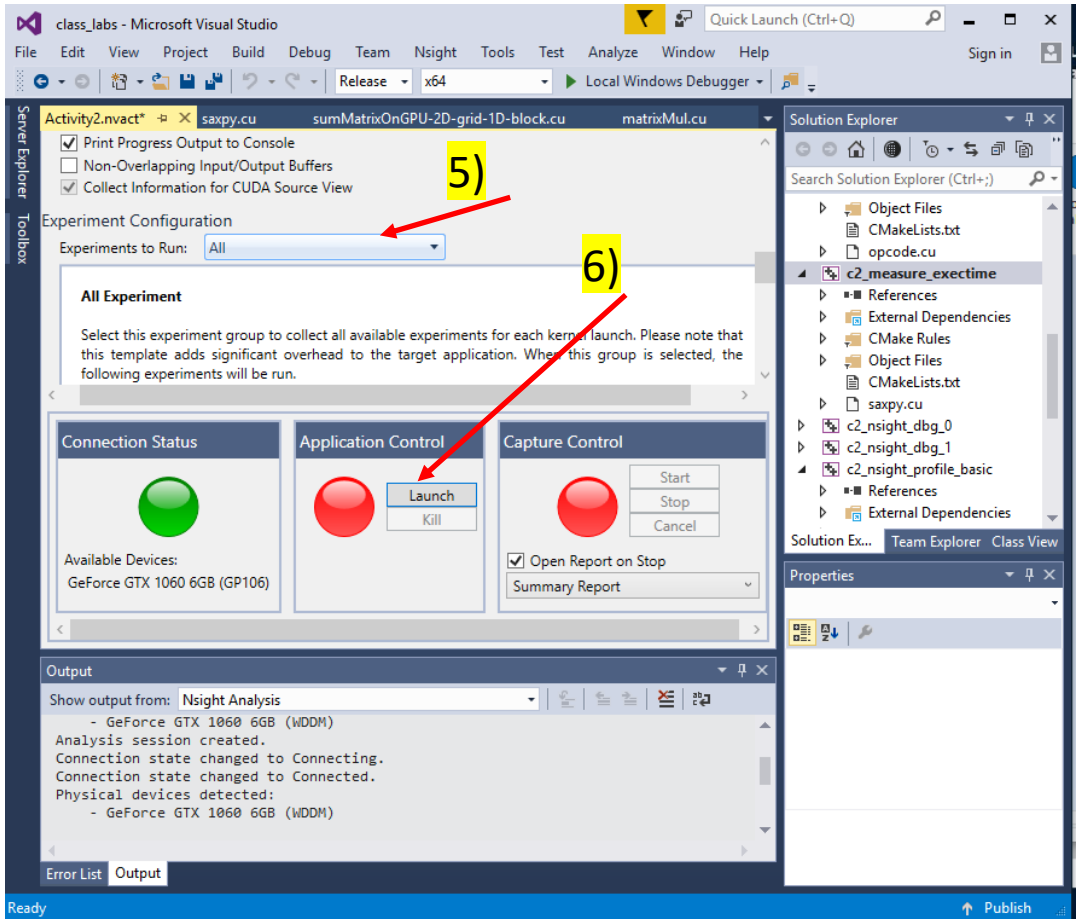
Measure execution time: CUDA profiler (1/3)

1) Class lab: c2_measure_exectime
2) Select “Release” mode
   Profiling should be done with “Release” mode
3) Nsight->Start Performance Analysis
4) Select “Profile CUDA Application”



Measure execution time: CUDA profiler (2/3)

5) Set Experiments to Run to “All”
6) Click “Launch”



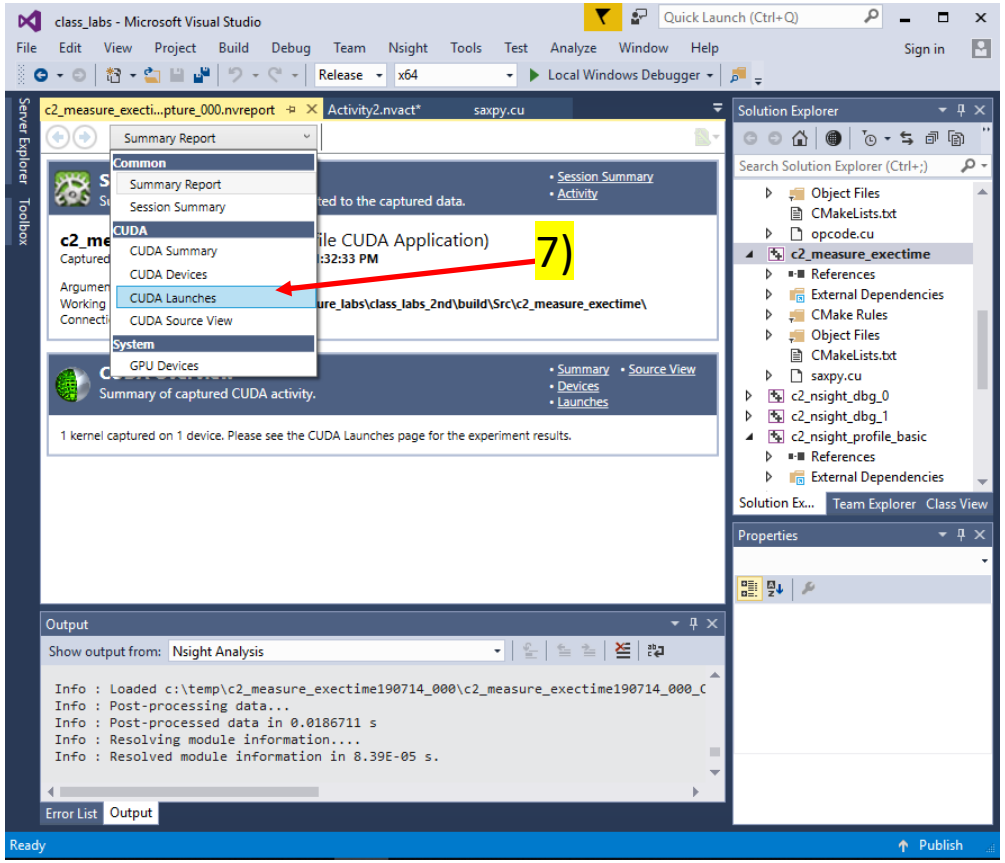
Measure execution time: CUDA profiler (3/3)

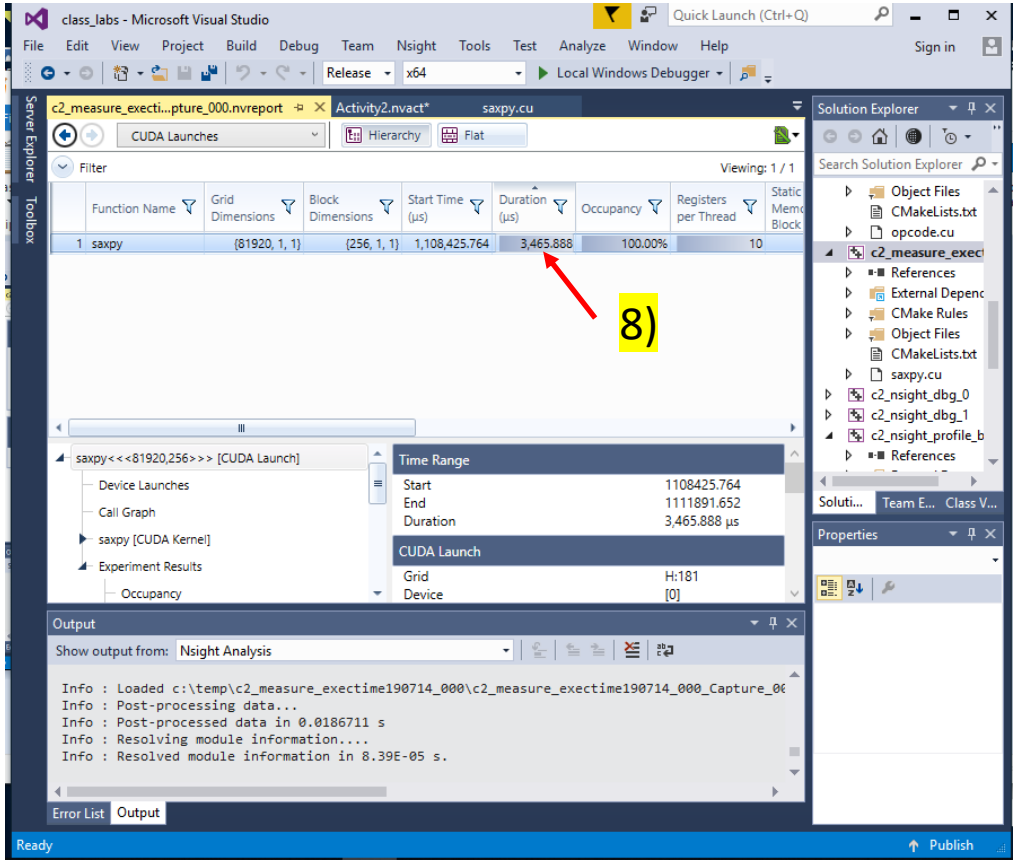
7) Scroll down “Summary Report” and Select “CUDA Launches”

8) Check “Duration” of kernel which you are interested in

“Timeline” ⇒ CPU/GPU co-operation

kernel execution time.





Only for execution time (duration)

only if error case

- We can also use "Trace Application"

system level = CPU / GPU at same time

1) Start performance analysis
2) Trace Application ⊕ Tools ~
3) Select "CUDA" under Trace Settings
4) Launch "Start"
5) Check "CUDA Launches" to view "Duration"

Measure execution time: Event method

1. Modify the class lab: c2_measure_exectime to measure execution time

```
cudaEvent_t start, stop;
cudaEventCreate(&start);
cudaEventCreate(&stop);

cudaEventRecord(start);
kernel<<< , >>>()
cudaEventRecord(stop);

cudaEventSynchronize(stop);
float milliseconds = 0;
cudaEventElapsedTime(&milliseconds, start, stop);
```