

Лекция 4

Инструменты профилирования:

- nvprof
- nvvp
- Nsight Compute CLI
- Nsight Compute

nvprof и Nsight Compute CLI

```
nvprof ./lab3c
```

Type	Time(%)	Time	Calls	Avg	Min	Max	Name
------	---------	------	-------	-----	-----	-----	------

GPU activities:

43.59%	2.1760us	1	2.1760us	2.1760us		
				2.1760us	gSum(int*, int*)	
41.67%	2.0800us	1	2.0800us	2.0800us		
				2.0800us	gInit(int*, int*)	
14.74%	736ns	1	736ns	736ns	736ns	[CUDA memcpy DtoH]
API calls:						
98.87%	131.54ms	2	65.772ms	6.9650us	131.54ms	cudaMalloc
.....						
0.09%	124.46us	2	62.229us	10.561us	113.90us	cudaFree
.....						
0.01%	14.599us	1	14.599us	14.599us	14.599us	
.....						
						cudaMemcpy

```
/Lecture3/Lab3-cuda-gdb # ncu --target-processes all ./lab3c
```

```
gInit(int *, int *), 2023-Feb-13 15:09:06, Context 1, Stream 7
```

```
Section: GPU Speed Of Light Throughput
```

```
-----  
DRAM Frequency cycle/nsecond          6.40  
SM Frequency    cycle/nsecond          1.29  
Elapsed Cycles          cycle          3,327  
Memory [%]              %              1.10  
DRAM Throughput          %              0.02  
Duration                usecond          2.56  
-----
```

WRN This kernel grid is too small to fill the available resources on this device, resulting in only 0.0 full waves across all SMs. Look at Launch Statistics for more details.

```
.....
```

```
/Lecture3/Lab3-cuda-gdb # ncu
```

```
--metrics gpu__time_duration.sum ./lab3c
```

```
gInit(int *, int *), 2023-Feb-13 18:42:52, Context 1, Stream 7
```

```
Section: Command line profiler metrics
```

```
-----
```

```
gpu  time duration.sum    usecond                29.50
```

```
-----
```

```
gSum(int *, int *), 2023-Feb-13 18:42:52, Context 1, Stream 7
```

```
Section: Command line profiler metrics
```

```
-----
```

```
gpu  time duration.sum    usecond                37.57
```

```
-----
```

```
/Lecture3/Lab3-cuda-gdb> nvprof --query-metrics  
===== Warning: Skipping profiling on device 0 since  
profiling is not supported on devices with compute  
capability 7.5 and higher.
```

Use NVIDIA Nsight Compute for GPU profiling and NVIDIA Nsight Systems for GPU tracing and CPU sampling.

Refer <https://developer.nvidia.com/tools-overview> for more details.

```
ip-011@linux-47dw:/home/malkov/WORKSHOP/PGP-2023>  
nvprof --query-metrics | less
```

Available Metrics:	Name	Description
Device 0 (GeForce GTX 1050):		
inst_per_warp:	Average number of instructions executed by each warp	
warp_execution_efficiency:	Ratio of the average active threads per warp to the maximum number of threads per warp supported on a multiprocessor	
.....		
gld_transactions_per_request:	Average number of global memory load transactions performed for each global memory load.	
gst_transactions_per_request:	Average number of global memory store transactions performed for each global memory store	
.....		

```
ip-011@linux-47dw:/home/malkov/WORKSHOP/PGP-2023>
```

```
nvprof -m gst_throughput ./lab3c
```

Invocations	Metric Name	Metric Description	Min	Max	Avg
Device "GeForce GTX 1050 (0)"					
Kernel: gSum(int*, int*)					
1	gst_throughput	Global Store Throughput	40.582MB/s	40.582MB/s	40.582MB/s
Kernel: gInit(int*, int*)					
1	gst_throughput	Global Store Throughput	71.303MB/s	71.303MB/s	71.302MB/s


```
/Lecture3/Lab3-cuda-gdb # ncu --list-sections
```

```
/Lecture3/Lab3-cuda-gdb # ncu --query-metrics
```

<https://docs.nvidia.com/nsight-compute/NsightComputeCli/index.html#nvprof-metric-collection>

```
/Lecture3/Lab3-cuda-gdb # ncu --metrics
```

```
lltex  t_bytes_pipe_lsu_mem_global_op_st.sum.per_second  
./lab3c
```

```
gInit(int *, int *), 2023-Feb-13 17:13:20, Context 1, Stream 7
```

```
Section: Command line profiler metrics
```

```
-----  
lltex  t bytes pipe lsu mem global op st.sum.per_second  
Mbyte/second                               89.89
```

```
-----  
gSum(int *, int *), 2023-Feb-13 17:13:20, Context 1, Stream 7
```

```
Section: Command line profiler metrics
```

```
-----  
lltex  t bytes pipe lsu mem global op st.sum.per_second  
Mbyte/second                               41.24  
-----
```

```
ip-011@linux-47dw:/home/malkov/WORKSHOP/PGP-2023>
```

```
nvprof -m gld_throughput ./lab3c
```

Invocations	Metric Name	Metric Description	Min	Max	Avg
Device "GeForce GTX 1050 (0)"					
	Kernel: gInit(int*, int*)				
1	gld_throughput	Global Load Throughput	0.0B/s	0.0B/s	0.0B/s
	Kernel: gSum(int*, int*)				
1	gld_throughput	Global Load Throughput	87.694MB/s	87.694MB/s	87.694MB/s

```
/Lecture3/Lab3-cuda-gdb # ncu --metrics
```

```
lltex  t_bytes_pipe_lsu_mem_global_op_ld.sum.per_second  
./lab3c
```

```
gInit(int *, int *), 2023-Feb-13 15:25:41, Context 1, Stream 7
```

```
Section: Command line profiler metrics
```

```
-----
```

```
lltex  t bytes pipe lsu mem global op ld.sum.per_second  
byte/second                                0
```

```
-----
```

```
gSum(int *, int *), 2023-Feb-13 15:25:41, Context 1, Stream 7
```

```
Section: Command line profiler metrics
```

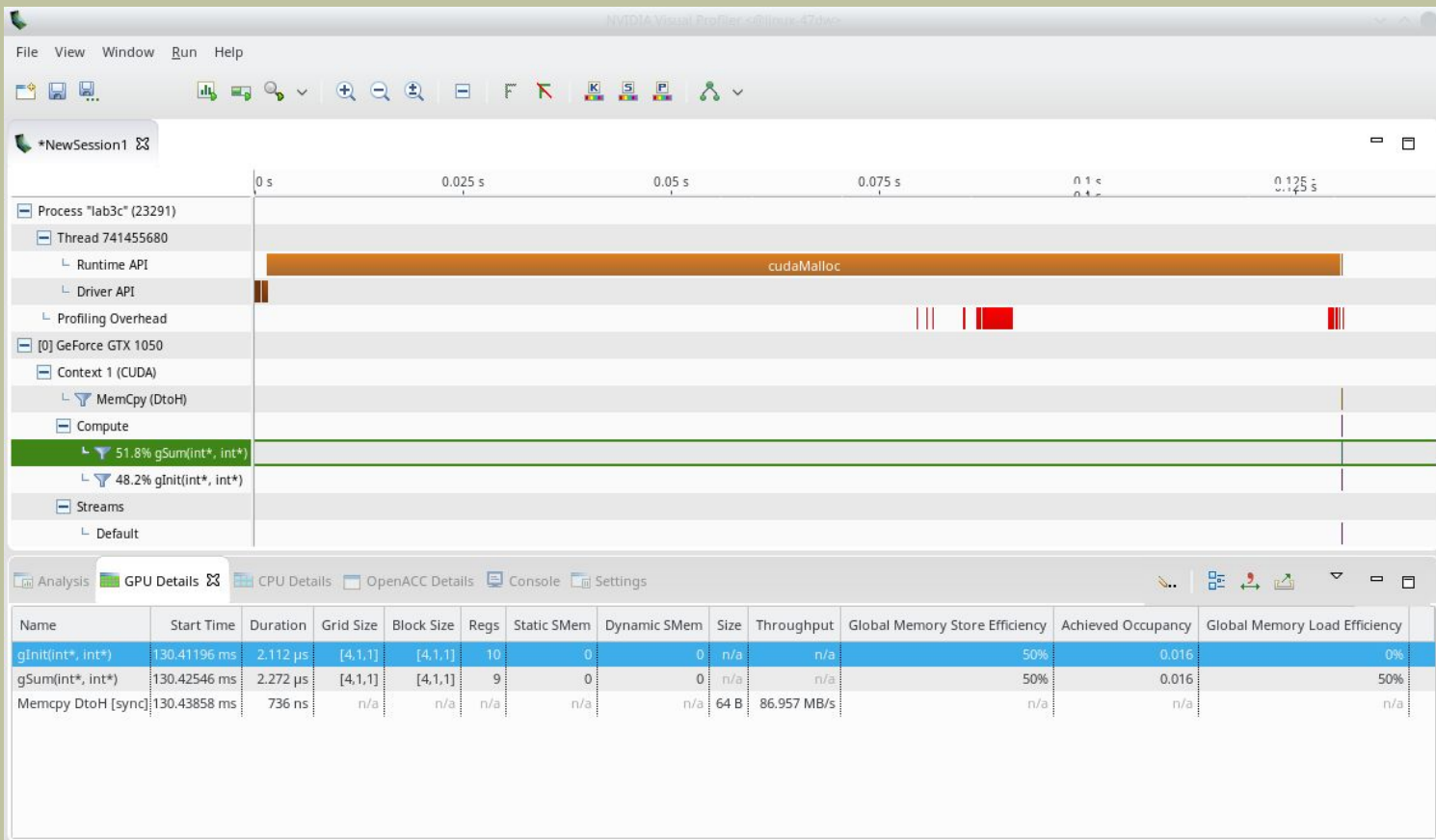
```
-----
```

```
lltex  t bytes pipe lsu mem global op ld.sum.per_second  
Mbyte/second                              82.47
```

```
-----
```

nvvp и Nsight Compute

```
ip-011@linux-47dw: /home/malkov/WORKSHOP/PGP-2023> nvvp ./lab3c
```



Metrics and Events <@linux-47dw>

Metrics and Events

Select metrics and events to be collected on individual devices

Device: [0] GeForce GTX 1050 ▾

Metrics

Events

☐ Device Memory Write Throughput

☐ Device Memory Write Transactions

☐ ECC Throughput

☐ ECC Transactions

☐ Global Load Throughput

☐ Global Load Transactions

☐ Global Load Transactions Per Request

☒ Global Memory Load Efficiency

☒ Global Memory Store Efficiency

☐ Global Store Throughput

☐ Global Store Transactions

☐ Global Store Transactions Per Request

☐ L2 Read Transactions

☐ L2 Write Transactions

Apply and Run

Cancel

OK

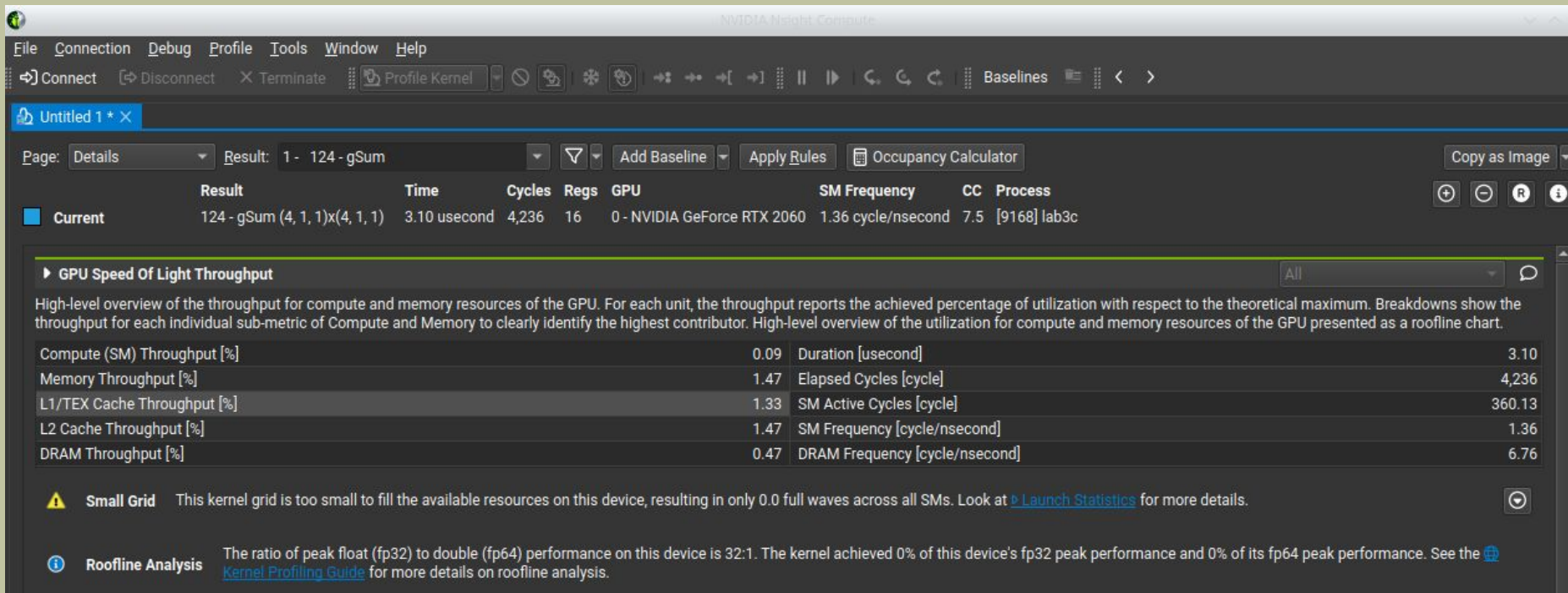
0.1 s0.125 s

Achieved Occupan

0.0160%

Open a dialog to configure metrics and events, and to run the application to collect

```
/Lecture3/Lab3-cuda-gdb # ncu-ui --target-processes all ./lab3c
```



The screenshot displays the NVIDIA Nsight Compute application window. The top menu bar includes File, Connection, Debug, Profile, Tools, Window, and Help. Below the menu is a toolbar with various icons for connecting, disconnecting, terminating, and profiling. The main window shows a summary of the current kernel's performance.

Page: Details **Result:** 1 - 124 - gSum **Add Baseline** **Apply Rules** **Occupancy Calculator** **Copy as Image**

	Result	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	124 - gSum (4, 1, 1)x(4, 1, 1)	3.10 usecond	4,236	16	0 - NVIDIA GeForce RTX 2060	1.36 cycle/nsecond	7.5	[9168] lab3c

GPU Speed Of Light Throughput

High-level overview of the throughput for compute and memory resources of the GPU. For each unit, the throughput reports the achieved percentage of utilization with respect to the theoretical maximum. Breakdowns show the throughput for each individual sub-metric of Compute and Memory to clearly identify the highest contributor. High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

Metric	Value	Unit
Compute (SM) Throughput [%]	0.09	Duration [usecond]
Memory Throughput [%]	1.47	Elapsed Cycles [cycle]
L1/TEX Cache Throughput [%]	1.33	SM Active Cycles [cycle]
L2 Cache Throughput [%]	1.47	SM Frequency [cycle/nsecond]
DRAM Throughput [%]	0.47	DRAM Frequency [cycle/nsecond]

Small Grid This kernel grid is too small to fill the available resources on this device, resulting in only 0.0 full waves across all SMs. Look at [Launch Statistics](#) for more details.

Roofline Analysis The ratio of peak float (fp32) to double (fp64) performance on this device is 32:1. The kernel achieved 0% of this device's fp32 peak performance and 0% of its fp64 peak performance. See the [Kernel Profiling Guide](#) for more details on roofline analysis.