

Profiling on Grace Hopper Jiri Kraus, Principal Devtech Compute | CASA Workshop JSC/Oct. 2024







- Nsight Compute: Fine-grained, kernel-level
- NVTX: Support and structure across tools
- Main purpose: Performance optimization But at their core, advanced measurement tools

The Nsight Suite Components How the pieces fit together





Nsight Compute

Detailed CUDA kernel performance

Optimize: GPU utilization, kernel implementation, memory access





- Multiple deployed versions may already exist, as profilers come...
 - ...with the CUDA Toolkit
 - ...with HPC SDK
 - ...standalone

- - Collect and Analyze via GUI
 - Collect on CLI, Analyze via GUI
 - Analysis GUI version >= collector version

Deployment and Setup https://docs.nvidia.com/nsight-systems/InstallationGuide/index.html

• Recommendation: Always install latest versions, especially for collection • All Nsight profilers available standalone - <u>https://developer.nvidia.com/nsight-systems/get-started</u> Via Website or Repositories, package managers (<u>https://developer.download.nvidia.com/devtools/repos/</u>) Compatibility: <u>https://developer.nvidia.com/nsight-systems/get-started#platforms</u>



Multiple deployed versions may already exist, as profilers come... ...with the CUDA Toolkit (2023.2.3 with CUDA 12 on JEDI):

👃 kraus1@jpblt-s01-01:~ \times + \sim

[kraus1@jpblt-s01-01 ~]\$ module load CUDA [kraus1@jpblt-s01-01 ~]\$ which nsys /p/software/jedi/stages/2024/software/CUDA/12/b [kraus1@jpblt-s01-01 ~]\$ nsys --version NVIDIA Nsight Systems version 2023.2.3.1001-328

• ...with HPC SDK (2024.1.1 with 24.3 on JEDI):



[kraus1@jpblt-s01-01 ~]\$ module load NVHPC/24.3-CUDA-12 [kraus1@jpblt-s01-01 ~]\$ which nsys [kraus1@jpblt-s01-01 ~]\$ nsys --version NVIDIA Nsight Systems version 2024.1.1.59-241133802077v0

...standalone (2024.4.1 on JEDI) – latest available 2024.6.1



Nsight Systems on JEDI

	—	×
in/nsys		
94139v0		



Call Stack, CPU Metric, and GPU Metric Sampling • Linux kernel setting required for call stack sampling (Linux Kernel Paranoid Level <= 2) and CPU metric

- sampling(Linux Kernel Paranoid Level <= 0):</pre>

 - on-linux
- profiling-tools

 - JUWELS-Booster slurm option: --disable-dcgm Disable DCGM Metrics collection. This is required for using profiling tools on the GPUs
 - JEDI: DCGM not running yet so not needed

• sudo sh -c 'echo kernel.perf_event_paranoid=2|0 > /etc/sysctl.d/local.conf' https://docs.nvidia.com/nsight-systems/InstallationGuide/index.html#requirements-for-x86-64-power-and-arm-sbsa-targets-

• Linux Kernel Paranoid Level = 2 default on JUWELS-Booster and JEDI

• Setting Linux Kernel Paranoid Level = 0 possible on JUWELS-Booster with slurm option: --disable-perfparanoid Setting Kernel not to be paranoid with respect to perf

• JEDI: Support will be added once ParTec software stack is integrated. If there is an urgent need contact jedi-adm@fz-juelich.de

 Some DCGM monitoring can interfere with GPU Metric Sampling, see https://docs.nvidia.com/datacenter/dcqm/latest/user-quide/feature-overview.html#concurrent-usage-of-nvidia-

DCGM and CUPTI are using the same hardware profiling capabilities

👃 kraus1@jpblt-s01-01:~ × + ~ [kraus1@jpbot-001-34 ~]\$ nsys status --environment Timestamp counter supported: Yes CPU Profiling Environment Check Root privilege: disabled Linux Kernel Paranoid Level = 2 Linux Distribution = RHEL Linux Kernel Version = 5.14.0-427.31.1.el9_4.aarch64+64k: OK Linux perf_event_open syscall available: OK Sampling trigger event available: OK Intel(c) Last Branch Record support: Not Available Kernel module: Not Available CPU Profiling Environment (process-tree): OK CPU Profiling Environment (system-wide): Fail including information on how to set the Linux Kernel Paranoid Level.

- Load Latest Nsight Systems
 - module load Nsight-Systems/2024.4.1
- Use the command line
 - ./jacobi -niter 10
- Inspect results: Open the report file in the GUI
 - Also possible to get details on command line
- - Useful to check validity of profile, identify important kernels

A First (I)Nsight Recording with the CLI

• srun nsys profile --trace=cuda,nvtx,mpi --force-overwrite=true --output=my_report.%q{SLURM_PROCID} \

• Either add --stats to profile command line, or: nsys stats --help

• Runs set of reports on command line, customizable (sqlite + Python):

Running [.../reports/gpukernsum.py jacobi_metrics_more-nvtx.0.sqlite]...

ces	Avg (ns)	Med (ns)	Min (ns)	Max (
20	1837518.0	1838466.5	622945	3055
2	11408.0	11408.0	7520	15

nsys stats --help-reports

- list all available reports
- Example: CUDA API sum, customize to show only "Stream" APIs
 - cp /opt/nvidia/nsight-systems/2024.1.1/host-linuxx64/reports/cuda_api_sum.py my_cuda_sum.py
 - Editing my_cuda_sum.py, for example:

 $\bullet \bullet \bullet$ 80 LEFT JOIN 81 StringIds AS ids ON ids.id == summary.nameId 82 WHERE Name LIKE "%Stream%" 84 ORDER BY 2 DESC •••

Running in the same way (reports from current folder picked up): • nsys stats -r my_cuda_sum --timeunit ms jacobi_2024.1-0.nsys-rep Processing [jacobi_2024.1-0.sqlite] with [./my_cuda_sum.py]...

** CUDA API Summary (my_cuda_sum):

Nan	StdDev (ms)	Max (ms)	Min (ms)	Med (ms)	Avg (ms)	Num Calls	Total Time (ms)	Time (%)
cudaStreamS	1.2879	3.1075	0.0023	0.3449	0.9530	40	38.1181	1.5
cuStreamCre	0.0941	0.4024	0.0018	0.0020	0.0255	18	0.4594	0.0
cudaStreamW	0.0012	0.0097	0.0006	0.0009	0.0012	80	0.0925	0.0
cudaStreamD	0.0098	0.0294	0.0043	0.0064	0.0108	6	0.0646	0.0
cudaStreamC	0.0128	0.0343	0.0024	0.0026	0.0082	6	0.0492	0.0
cuStreamDes	0.0163	0.0344	0.0113	0.0229	0.0229	2	0.0457	0.0
cuStreamSvr	0.0004	0.0029	0.0022	0.0025	0.0025	4	0.0101	0.0

Example: Available reports Extending and customizing

The following built-in reports are available: cuda_api_gpu_sum[:nvtx-name][:basel:mangled] -- CUDA Summary (API/Kernels/MemOps) cuda_api_sum -- CUDA API Summary cuda_api_trace -- CUDA API Trace cuda_gpu_kern_gb_sum[:nvtx-name][:basel:mangled] -- CUDA GPU Kernel/Grid/Block Summary cuda_gpu_kern_sum[:nvtx-name][:basel:mangled] -- CUDA GPU Kernel Summary cuda_gpu_mem_size_sum -- CUDA GPU MemOps Summary (by Size) cuda_gpu_mem_time_sum -- CUDA GPU MemOps Summary (by Time) cuda_gpu_sum[:nvtx-name][:basel:mangled] -- CUDA GPU Summary (Kernels/MemOps) cuda_gpu_trace[:nvtx-name][:basel:mangled] -- CUDA GPU Trace cuda_kern_exec_sum[:nvtx-name][:basel:mangled] -- CUDA Kernel Launch & Exec Time Summary cuda_kern_exec_trace[:nvtx-name][:basel:mangled] -- CUDA Kernel Launch & Exec Time Trace dx11_pix_sum -- DX11 PIX Range Summary dx12_gpu_marker_sum -- DX12 GPU Command List PIX Ranges Summary dx12_pix_sum -- DX12 PIX Range Summary mpi_event_sum -- MPI Event Summary mpi_event_trace -- MPI Event Trace network_congestion[:ticks_threshold=<ticks_per_ms>] -- Network Devices Congestion nvtx_gpu_proj_sum -- NVTX GPU Projection Summary nvtx_gpu_proj_trace -- NVTX GPU Projection Trace nvtx_kern_sum[:basel:mangled] -- NVTX Range Kernel Summary nvtx_pushpop_sum -- NVTX Push/Pop Range Summary nvtx_pushpop_trace -- NVTX Push/Pop Range Trace nvtx_startend_sum -- NVTX Start/End Range Summary nvtx_sum -- NVTX Range Summary nvvideo_api_sum -- NvVideo API Summary openacc_sum -- OpenACC Summary opengl_khr_gpu_range_sum -- OpenGL KHR_debug GPU Range Summary opengl_khr_range_sum -- OpenGL KHR_debug Range Summary openmp_sum -- OpenMP Summary osrt_sum -- OS Runtime Summary um_cpu_page_faults_sum -- Unified Memory CPU Page Faults Summary um_sum[:rows=<limit>] -- Unified Memory Analysis Summary um_total_sum -- Unified Memory Totals Summary vulkan_api_sum -- Vulkan API Summary vulkan_api_trace -- Vulkan API Trace vulkan_gpu_marker_sum -- Vulkan GPU Range Summary vulkan_marker_sum -- Vulkan Range Summary wddm_queue_sum -- WDDM Queue Utilization Summary

For more information, use '--help-reports <report_name>'

ne

Synchronize eate WaitEvent Destroy Create stroy_v2 ynchronize

System-level Profiling with Nsight Systems

- Global timeline view
 - CUDA HW: streams, kernels, memory
- Different traces, e.g. CUDA, MPI
 - correlations API <-> HW
- Stack samples
 - bottom-up, top-down for CPU code
- (GPU metrics)
- Events View
- looks at single process (tree)

								×
			I	-	<u> </u>	rning	s, 16 me	ssages
950ms	2s	+	50ms			+100	ms	
	i · · ·					Ī	-	
н					1	_	hil.	1
н								
	1							
							M.	
	P. A.				4			
(MPI_Sendr	ecv [41,692 MPI_S M	MPI.	_S MPI_S	sen MPI	MP	Pl	S)
mcpy		c		1			<u>M</u>	
								-
	-	-						•
		Name		•				Q
	GPU	Context		Descri	ption:			
	GPU 0	Stream 13		void i	acobi k	ernel	<(int)32	, 🔺
	GPU 0	Stream 13		(int)3	2>(floa	t *, co	onst floa	at
	GPU 0	Stream 14		*, floa	i t *, int, s: 1.882'	int, i ५१९	nt, bool)
	GPU 0	Stream 13	_	Ends:	1,88565	s (+3,	056 ms)	-

8

Correlating Events on the Timeline Selecting events in one row highlights related events

	+283ms
I	

- - Kernel launch on host returns
 - Kernel runs on GPU
- Visualized in profiler

+282,48ms	+282	2,49ms	+282,5ms
obi_kernel			cudaEventRecord

CUDA's execution model is asynchronous

2s 🔻	+	282,47ms	+282,48ms	+282	,49ms	+28
CUDA HW (0000:03:00.0 - NVIDIA A100-					7	
[All Streams]					7	void jacobi_kernel
▼ 39.0% Kernels					7	void jacobi_kernel
 >99.9% jacobi_kernel 					7	void jacobi_kernel
100.0% void jacobi_kernel(float						void jacobi_kernel
<0.1% initialize_boundaries						
61.0% Memory						
60.6% Default stream 7					7	
31.2% Stream 14						void jacobi_kernel
7.9% Stream 17						
5 streams hidden — +						
Threads (9)						
✓ [10232] MPI Rank 0 -						
MPI	7	7				
CUDA API			jacobi_kernel			
Profiler overhead						

Correlating Events on the Timeline Selecting events in one row highlights related events

		_	
	Options		▲ <u>3 warnings, 15 massages</u>
3s -	ns +485.96ms +485.	98ms +486ms +486.02ms +486.04m	ns +486.06ms +486.02m 📤
CPU (288)	100% 0		
GPU (0009:01:00.0 - NVIDIA GH20			
CUDA HW (0009:01:00.0 - NVIDIA	Kernel Memory		
▶ 36.7% Kernels		copy(float *, float *, int, int)	
▶ 63.3% Memory			
 Threads (8) 			
▼ ▼ [17780] transpose +	0 to 100%	7	
CUDA API	01100001100		
Profiler overhead			
✓ [17797] transpose -	0 to 100%		3s - +4
6 threads hidden 🗕 🕂	0 to 100%		▶ CPU (288) 100% 0
			GPU (0009:01:00.0 - NVIDIA GH20
	4		CUDA HW (0009:01:00 0 - NVIDIA Kernel

Ξ Timeline View ▼	options	
3s -		+490
CPU (288)	100% 0	
GPU (0009:01:00.0 - NVIDIA GH20		
CUDA HW (0009:01:00.0 - NVIDIA	Kernel Memory	
▶ 36.7% Kernels		-0-04
▶ 63.3% Memory		
 Threads (8) 		
▼ ▼ [17780] transpose →	0 to 100%	
CUDA API		
Profiler overhead		
✓ [17797] transpose +	0 to 100%	
6 threads hidden — 🕂	0 to 100%	
		4

#	Name	Start	Duration	GPU	Context
1	initialize_boundaries(float *, float *, float, int, int, int, i	1,88146s	15,360 μs	GPU 0	Stream 7
2	void jacobi_kernel<(int)32, (int)32>(float *, const floa	1,88259s	3,056 ms	GPU 0	Stream 13
3	void jacobi_kernel<(int)32, (int)32>(float *, const floa	1,88574s	3,052 ms	GPU 0	Stream 13
4	void jacobi_kernel<(int)32, (int)32>(float *, const floa	1,88884s	3,051 ms	GPU 0	Stream 13
5	void jacobi_kernel<(int)32, (int)32>(float *, const floa	1,89193s	3,052 ms	GPU 0	Stream 13
6	void iscobi kornel «(int)?? (int)??» (fleet * const flee	1.005020	2.051 mg	CDU A	Ctroom 1

Nsight Systems Basic Workflow Navigating the timeline and finding interesting areas

			×
1 1 1 1 4	1 2 warnin	gs, 16 me	<u>ssages</u>
2,5s	s		3,5 📥
			_
			<i></i>
	Host		
Ucudaria	ici iost		
			_
			-
Name -			9
Description:			
3			
3			
3			
3			
· ▼			

• #include "nvtx3/nvToolsExt.h"

- NVTX v3 is header-only, needs just -ldl
- C++ and Python APIs
- Fortran: NVHPC compilers include module
 - Just use nvtx and -lnvhpcwrapnvtx
 - Other compilers: See blog posts linked below
- **Definitely: Include** PUSH/POP macros (see links below)

PUSH RANGE (name, color idx)

- Sprinkle them strategically through code
 - Use hierarchically: Nest ranges
- Not shown: Advanced usage (domains, ...)
- Similar range-based annotations exist for other tools
- e.g. <u>SCOREP_USER_REGION_BEGIN</u>

https://developer.nvidia.com/blog/customize-cuda-fortran-profiling-nvtx/

Adding NVTX Simple range-based API

- https://github.com/NVIDIA/NVTX and https://nvidia.github.io/NVTX/#how-do-i-use-nvtx-in-my-code
- https://developer.nvidia.com/blog/cuda-pro-tip-generate-custom-application-profile-timelines-nvtx/


```
int main(int argc, char** argv) {
```

```
PUSH_RANGE("computation", 2)
jacobi_kernel<<</* ... */, compute_stream>>>(...);
cudaStreamSynchronize(compute_stream);
```


- Same section of timeline as before Events view: Quick navigation
- Like manual timing, only less work
- Nesting
- Correlation, filtering

Adding some Color Code annotation with NVTX

	2s -	+350m	5 +	400ms	+450ms	+500ms	+550ms	+600n	ns	+650ms	+700ms
CPU (96)											
GPU (0000:03:0)	0.0 - NVIDIA										
CUDA HW (000)	0:03:00.0 🧎					II		ΙI			
 Threads (8) 											
🕶 🗹 [17825] MP	Pl Rank 0 🝷										
MPI						MPI_Sendrec.	MPI_Sendrecv [63,6	95 ms] [MPI_Sendre	ecv [55,34	44)	
NVTX			si Jacob	i <mark>ngle_gpu [93,3</mark> pi solv	48 ms]	init [2,490 s]	MPI_Warr	nup [178,273 ms]			
CUDA API		cudaHostAllo	c	cud	aMemcpy C) c)	culpcOpenMemH	andle		, Cu.	
Profiler over	head										
🗸 [17878] jac	obi 🝷										
6 threads hidd	den — +	-	-	-				-			
		•									Þ
Events View	-										
								Name	*		
#	Name			Start	Duratio	n Tl	D	Category		Description:	
1	🕨 📘 init			0,17077s	2,490 s	1	7825			it_000	
5	🔻 🛛 Jacol	bi solve	Zoom to	o Selected o	n Timeline	ns 1	7825			Begins: 2,66035s	; ±017 263 µc)
6	🔹 🔤 it	t_000	Copy Se	elected	.6	3 μs 1	7825			Thread: 17825	+517,205 μ3)
7		kernel		2,66035s	682,25	2 μs 1	7825				
8		MPI		2,66103s	105,39	1 μs 1	7825				
9		norm		2,66114s	128,15) μs 1	7825				
10	🕨 🕨 📘 it	t_001		2,66127s	751,18	2 μs 1	7825		-		

Nsight Systems Workflow with NVTX Repeating the analysis

6	NV	IDIA	Nsight (1997)	Syste	ms	2021
<u>F</u> ile	e <u>v</u>	/iew	<u>T</u> ools	<u>H</u> el	р	
jac	obi_	_metr	rics_mo	re-nv	tx.0	.nsys
	⊒ T	imeli	ne <mark>Vi</mark> ev	v		-
						2s -
Þ	СР	U (96	5)			_
Þ	GP	U (00	00:03:0	0.0 -	NV	IDIA
►	CU	DAH	I <mark>W (00</mark> 0	00:03:	00.0) - N
-	Th	reads	(8)			
	- V	[17	825] M	PI Rar	n <mark>k (</mark>) -
		MPI				
		NIV/T				
		INVE	~			
		CUD.	A API			
		Profi	ler ove	rhead		
	V	[17	878] jao	obi -	•	
	6	threa	ads hid	den	-	• +
E	ven	ts Vie	w	Ŧ	-	
#				Nar	ne	
	1			•		nit
	5			•		lacol
	6				-	it
	7	7				
	8	3				
	9)				
	1	0			۲	it
	1.	4			•	it
	1:	8			•	l it
	2	2			•	it

1.4.1

rep ×						
]				2	⊠ ∈	<mark>२</mark> 1
+660,4ms +660,5ms	2s 660,6070ms	+660,7ms	+660,8ms	+660,9ms		
		1				
			Jacobi solve [7,810 m	s]		
			it_000 [917,263 μs]			
	k	ernel [682,252 µs]				
c cuda		cudaEventSynchron	ize			
	2					

	Start	Duration	TID	Cate
	0,17077s	2, <mark>4</mark> 90 s	17825	
i solve	2,66034s	7,810 ms	17825	
_000	2,66035s	917,263 μs	17825	
kernel	2,66035s	682,252 μs	17825	
MPI	2,66103s	105,391 μs	17825	
norm	2,66114s	128,150 μs	17825	
001	2,66127s	751,182 μs	17825	
002	2,66202s	767,433 μs	17825	
003	2,66279s	752,632 μs	17825	
004	2,66354s	762,572 μs	17825	

				—		\times
x ↓	s	+661,1	<u>∧</u> 2 ms	<u>warning</u> +661,	<u>is, 16 me</u> : 2ms	ssages
			r			
	MPI_Se	ndrecv []	MPI	MPI_Allre	duce [
	MP	[105,391 μ	5	norm [128	3,150 µs]	
						•
2004	Name	• Descrin)			٩
		it_000 Begins: Ends: 2 Thread	: 2,6603 ,66127s : 17825	5s (+917,26	53 µs)	
						P

Only profile what you need – all profilers have some overhead • Example: Event that occurs after long-running setup phase

- Bonus: lower number of events leads to smaller file size
- Add to nsys command line:
 - --capture-range=nvtx --nvtx-capture=any_nvtx_marker_name \
 --env-var=NSYS_NVTX_PROFILER_REGISTER_ONLY=0 --kill none
 - Use <u>NVTX registered strings</u> for best performance
- Alternatively: cudaProfilerStart() and -Stop()
 - --capture-range=cudaProfilerApi

Minimizing Profile Size Shorter time, smaller files = quicker progress

GPU Metric Sampling

- Use nsys profile ... --gpu-metrics-device= ...
- With multiple GPUs per node --gpu-metrics-device needs to be consistent with application GPU affinity

export CUDA_VISIBLE_DEVICES=\${SLURM_LOCALID} nsys profile ... --gpu-metricsdevice=\${SLURM_LOCALID} ./a.out

NVIDIA Nsight Systems 20
<u>File View T</u> ools <u>H</u> elp
matrixMulCUBLAS.nsys-rep
■ Timeline View
▶ CPU (288)
✓ GPU (0009:01:00.0 - NVI
 GPU Metrics [10 kHz]
GPC Clock Frequenc
SYS Clock Frequency
GR Active
SMs Active
 SM Instructions
SM Issue
Tonsor Activo
SM Warp Occupancy
v ow warp occupancy
DRAM Bandwidth
DRAM Read Bandy
DRAM Write Pandy
NVI ink PV Pandwidt
NVLink TX Dandwidt
NVLINK TX Bandwidt DOIs Desiduidth
PCle Bandwidtn
 CUDA HW (0009:01:00.0
 30.7% Kernels 21.5% transposeMain
 13.1% transposeDiac
▶ 11.4% transposeCoa
▶ 11.1% copySharedM
11.0% transposeNoE
11.0% transposeCoa
▶ 10.8% transposeFine
✓ 10.0% copy 100.0% copy/float
L 62.2% Momony
 Threads (8)
 ✓ [17780] transpose
- CUDA API
CUDA Overhead
Profiler overhead
✓ [17797] transpose
6 threads hidden
s on ease madel

24.5.1		$ \Box$ \times
_		
< transp	ose.nsys-rep	
▼ © (Options	📼 Q 🖵 📩 1x 🔥 <u>3 warnings, 15 messages</u>
3s -	100%	+490ms 3s 496.70ms 0ms +510ms +520ms +530ms
A GH20	U	
	memory	
	0 to 2.00 GHz	
	0 to 1.6/ GHZ	
	0 to 100%	
	100%	
	0	
	0 to 100%	
idth	0 to 100%	DRAM Bandwidth Throughput %
idth	0 to 100%	DRAM Read Bandwidth 20.9
1	0 to 100%	
	0 to 100%	
	U to TUU% Kernel	
	Memory	
е		
onal		
esced		
m ankConfli		
seGraine		
Grained		
, float *, ir		
•	0 to 100%	
		cudaEventsynchronize
ي ا		
	0 to 100%	
-+	0 to 100%	
		4 A

Unified Memory Profiling nsys profile ... --cuda-um-cpu-page-faults true --cuda-um-gpu-page-faults true

Fault Based Migrations

😸 NVIDIA Nsight Systems	2024.5.1						
<u>File View Tools Help</u>							
test.um_events.managed.a	accesscounters.lego	-cg1-qs-91.D	riverVersion.550.54	.14.202404	08130441.ns	sys-rep 🗙 me	mloc_test.u
Timeline View	▼ Ø Options						
	0s •	j	0ms +680	ms	+700ms	+720n 7	27.3ms +7
▶ CPU (72)		100% 0					
- CUDA HW (0009:01:00	.0 - NVIDIA GH200	Kernel					
▶ 89.5% Context 1		memory					
 10.5% Unified memory 	ry						
GPU Page Faults							
✓ 100.0% Memory							UM GPU
63.6% HtoD trans	fer						Ends: 0.1
36.4% DtoH trans	fer						Virtual a
 Threads (8) 							Memory
▼ √ [1798] memioc_te	est •	0 to 100%					
NVTX							UVM Eve
CUDA API							
Profiler overhead	2						
✓ [1812] memloc_te	est •	0 to 100%					
Profiler overhead							
6 threads hidden	-+	0 to 100%					
			4				
Events View 🔻							
# Aname					S	tart	Duration
3	UVM Events: GF	PU first touch			(0.660564s	178.886
4	UVM Events: Pre	efetch to GP	U			0.839494s	24.270 r
1 5	UVM Events: Pre	efetch to CP	U			0.863765s	11.855 r
6	UVM Events: Mi	grate to GPU]			0.875621s	103.567

— 🗆 ts.managed.lego-cg1-qs-91.DriverVersion.550.54.14.20240408130441.nsys-rep Ξ Q 1x (i) 16 messages +760ms +780ms +800ms +840 +820ms page fault 0.726715s 728354s (+1.640 ms) ddress: 0xfffaf8d0 of page faults: 40 access type: Write first touch [178.886 ms] 1111 . . . - F -Name - Description: Category TID UVM Events: GPU first touch 1798 Begins: 0.660564s 1798 ms Ends: 0.83945s (+178.886 ms) Thread: 1798 1798 ms 1798 7 ms \Box

Sila Miaw Taala	ystems 2024.5.1	
File View Tools	Help	
memloc_test.um_ev	/ents.managed.accesscou	inters.lego-cg1-
Timeline View	 ▼ Ø Options 	
	0s •	20
 CPU (72) 		100%
	0-01-00 0 NIV/IDIA OLIO00	0 Kernel
✓ CUDA HW (000)	9:01:00.0 - NVIDIA GH200	Memory
 87.9% Contex 12.1% Unified 	memory	
GPU Page F	aults	(
	nory	
86.6% Htc	D transfer	
13.4% Dto	H transfer	
✓ Threads (8)		
👻 🗸 [1920] mer	mloc_test •	0 to 100%
NVTX		
CUDA API		l l
Profiler over	rhead 🖌	
👻 🗸 [1934] mer	mloc_test •	0 to 100%
Profiler over	rhead	
6 threads hide	den — 🕂	0 to 100%
		4
Events View	•	
#	Name	
 3	UVM Events: G	PU first touch
4	UVM Events: P	refetch to GPU
II 5	UVM Events: P	refetch to CPU
6	UVM Events: N	ligrate to GPU

41

Access Counter Based Migration

Access Count	er Ba	sed M	ligrat	ion	Check D.
					rofilipo
1-qs-91.DriverVersion.550.54.14.2024040813	0441.nsys-rep ×	memloc_test.um_e	events.managed.le	go-cg1-qs-91.Drive	rVersion.550.54.14.20240408130441.nsy
				Ξ Q	1x (i) <u>16 messages</u>
'20ms +730ms +740ms +	750ms 756.1ms 0	ms +770ms	+780ms	+790ms	+800ms +810ms +820ms
Writ Wr					
	W				
	UM G Begin	PU page fault s: 0.755314s			
	Ends: Virtua	0.757764s (+2.450 I address: 0xfffc30	ms) 680000		
	Numb Memo	er of page faults: 3 ory access type: Wri	1 ite		
	U۱	UVM Events [8 /M Events: GPU first	19.710 ms] touch [104.486 ms]		
		cudaMem	set		C
4					▼ ▶
				Name	▼<
h	Start 0 719785s	Duration	TID 1920	Category	Description:
U.	0.824302s	25.339 ms	1920		Begins: 0.719785s Ends: 0.824271s (+104.486 ms)
U	0.849643s	12.347 ms	1920		Thread: 1920

80.117 ms

1920

Ŧ

•

0.861992s

Ģ

Unified Memory Profiling nsys profile ... --cuda-um-cpu-page-faults true --cuda-um-gpu-page-faults true

Fault Based Migrations

NVIDIA Nsight Systems 2024.5.1				
<u>File View Tools Help</u>				
test.um_events.managed.accesscounters.lego	-cg1-qs-91.D	riverVersion.550.54.14.202404	108130441.nsys-rep × n	nemloc_test.u
■ Timeline View ▼ @ Options				
0s -		+840ms	+845 845.98ms	+85
▶ CPU (72)	100% 0			
- CUDA HW (0009:01:00.0 - NVIDIA GH200	Kernel Memory			
▶ 89.5% Context 1				
 10.5% Unified memory 				
GPU Page Faults				
 100.0% Memory 		цынаралан	and a state of	de la de
63.6% HtoD transfer		Libert de Belle de Let	hininininininini	skitelinit, eled
36.4% DtoH transfer			Begins: 0.8	45983s
 Threads (8) 			Ends: 0.845	993s (+9.824
✓ [1798] memloc_test .	0 to 100%		Migration c	ause: User pre 198 812 GiB
10.000			Virtual addr	ess: 0xfffb294
NVIX				UVM Eve
► CUDA API		cudaMemPrefetchAs	cudaMemPrefetchA	cudaMem
Profiler overhead				
✓ [1812] memloc_test ◆	0 to 100%			
Profiler overhead				
6 threads hidden+	0 to 100%			
		4		
		1		
Events View 👻				
# AName			Start	Duration
3 UVM Events: GF	PU first touch		0.660564s	178.886
4 UVM Events: Pro	efetch to GPl	J	0.839494s	24.270 ו
5 UVM Events: Pr	efetch to CPl	J	0.863765s	11.855
6 UVM Events: Mi	grate to GPU		0.875621s	103.567

							-	-		Х
n_events	s.mana	ged.lego-	cg1-qs-91.Driv	erVersion	.550.54.14.	2024040	0813044	1.nsys-r	ep ×	
				📼 6			1x	(j) <u>16</u>	messa	iges
)ms			+855ms			+860m	IS			-
										- 1
		Hereit		Constant	an Bhana					-
										- 1
		1111								- 1
dhallalti			litebalaalia.	Linduli			.thibibilin	inaganya		-
µs)										- 1
ytes fetch	T							•		
S		1	1 1	1	I	1	1	• •	1	
nts: Prefet	tch to G	915 msj :PH [24 270) msl							
PrefetchA		cudaMer	nPrefetchA	cuda	MemPrefetcl	hA)	cudaM	emPrefe	tchA	_
	T									
								•		
										_
										-
									Þ	-
										_
				Manar						
				Name	Descript	ion:				<u> </u>
ms	110	1	Category	-		onte: Pr	afotob to	GPU		
ns	1799	,			Begins:	0.839494	4s			
ns	1798				Ends: 0. Thread:	863764s 1798	(+24.27	u ms)		
ms	1798	}								
		• 								
									(

😸 NVIDIA Nsight Systems 2024.5.1	
<u>File View T</u> ools <u>H</u> elp	
memloc_test.um_events.managed.accesscou	nters.lego-cg1-
■ Timeline View ▼ @ Options.	
0s •	+
▶ CPU (72)	100%
	0
 CUDA HW (0009:01:00.0 - NVIDIA GH200 	Memory
▶ 87.9% Context 1	
 12.1% Unified memory 	
GPU Page Faults	
86.6% HtoD transfer	
13.4% DtoH transfer	
 Threads (8) 	
▼ 【1920] memloc_test ●	0 to 100%
NVTX	
► CUDA API	
Profiler overhead	
-	0 to 100%
▼ 【1934] memloc_test ◆	<u>i</u>
Profiler overnead	
6 threads hidden+	0 to 100%
	4
Events View	
Lvents view	
# A Name	
3 UVM Events: G	PU first touch
4 UVM Events: P	refetch to GPU
5 UVM Events: P	refetch to CPU
6 UVM Events: M	ligrate to GPU
4	grate to or o

Access Counter Based Migration

cess Co	ounter Ba	ased N	<i>ligratic</i>	on 🤇	Check	Profil.
1.DriverVersion.550.54.14	.20240408130441.nsys-rep	× memloc_test.um	_events.managed.lego-(cg1-qs-91.Driver	Version.550.54.1	4.20240408130441.nsy
ims	+830ms	+{835.48ms	+840ms	Ξ Q	+845ms	1x (i) <u>16 messages</u>
				-		
			a da da da da da da da da			
adadadada kalendari ata	alahahahahahahahaha	• HtoD to Time: 0.1	femory operations in s: transfer 83553s	inda	thathathathat	.kilkd.antuta.kintut
1 1 1	1 1 1	IIVM Events	1 1 1	1 1	1 1	· · · · ·
		LIVM Events: Prefetch	[819.710 ms] h to GPU [25.339 ms]			The second secon
cudaMemPrefetchA	cudaMemPrefetchA	UVM Events: Prefetch	[819.710 ms] h to GPU [25.339 ms]] cudaMemPrefet	ich	MemPrefetch)	cudaMemPrefetch
cudaMemPrefetchA	cudaMemPrefetchA	UVM Events: Prefetch	[819.710 ms] h to GPU [25.339 ms]] cudaMemPrefet	ich) Cuda	MemPrefetch)	cudaMemPrefetch
cudaMemPrefetchA	cudaMemPrefetchA	UVM Events: PrefetchA.	[819.710 ms] h to GPU [25.339 ms]] cudaMemPrefet	cuda	MemPrefetch)	cudaMemPrefetch
cudaMemPrefetchA	cudaMemPrefetchA	UVM Events: PrefetchA	[819.710 ms] h to GPU [25.339 ms]] cudaMemPrefet	cuda	MemPrefetch)	cudaMemPrefetch
cudaMemPrefetchA)	cudaMemPrefetchA	UVM Events: PrefetchA.	[819.710 ms] h to GPU [25.339 ms]] cudaMemPrefet	ich) cuda	MemPrefetch)	cudaMemPrefetch
cudaMemPrefetchA)	cudaMemPrefetchA	UVM Events: PrefetchA.	[819.710 ms] h to GPU [25.339 ms]] CudaMemPrefet	cuda	MemPrefetch	CudaMemPrefetch
CudaMemPrefetchA)	cudaMemPrefetchA	UVM Events: PrefetchA.	[819.710 ms] h to GPU [25.339 ms]] CudaMemPrefet	Ich) Cuda	MemPrefetch)	
cudaMemPrefetchA		UVM Events: PrefetchA	[819.710 ms] h to GPU [25.339 ms]] CudaMemPrefet	Ich) Cuda	MemPrefetch)	
cudaMemPrefetchA	CudaMemPrefetchA	UVM Events: PrefetchA	[819.710 ms] h to GPU [25.339 ms] 	tegory	MemPrefetch)	
	cudaMemPrefetchA Start 0.719785s	UVM Events: PrefetchA.	[819.710 ms] h to GPU [25.339 ms]] CudaMemPrefet 1 1 1 1 1 1 1 1 1 1 1 1 1	tegory	MemPrefetch	cudaMemPrefetch
	cudaMemPrefetchA)	UVM Events: PrefetchA.	[819.710 ms] h to GPU [25.339 ms] 	tegory	MemPrefetch Description: UVM Events: P Begins: 0.82430 Ends: 0.849642 These 5: 1000	cudaMemPrefetch
CudaMemPrefetchA)	CudaMemPrefetchA)	UVM Events: PrefetchA. CudaMemPrefetchA.	[819.710 ms] h to GPU [25.339 ms] 	tegory	MemPrefetch Description: UVM Events: P Begins: 0.82430 Ends: 0.849642 Thread: 1920	cudaMemPrefetch

18

Q

Unified Memory Profiling nsys profile ... --cuda-um-cpu-page-faults true --cuda-um-gpu-page-faults true

Fault Based Migrations

🗮 NVIDIA Nsight S	ystems 2024.5.1						
<u>File View T</u> ools	<u>H</u> elp					_	
<pre>{test.um_events.ma</pre>	naged.accesscounters.lego	-cg1-qs-91.D	riverVersion.550).54.14.202404	08130441.ns	ys-rep 🗙 me	mloc_test.u
Timeline View	v ▼ @ Options						
	0s 🕶		+880ms	+890ms	+900ms	+910ms	920.
 CPU (72) 		100%					
- CUDA HW (000	9:01:00.0 - NVIDIA GH200	Kernel Memory					
▶ 89.5% Contex	dt 1	-					
	I memory						
GPU Page F	aults		Read Rea	Wri Writ			
	mory				-	Li du autoratio	
63.6% Hte	oD transfer				andradisantasada	հետուստետնես	nataran di dan marana
36.4% Dto	oH transfer						
 Threads (8) 							
👻 🗸 [1798] me	mloc_test •	0 to 100%					
NVTX							UVM Eve
CUDA API							
Profiler ove	rhead 🖉						
👻 🗸 [1812] me	mloc_test •	0 to 100%			<u> </u>		
Profiler ove	rhead						
6 threads hid	den+	0 to 100%					
			4				
Events View	•						
#	Name				St	art	Duration
3	UVM Events: GF	PU first touch			0	.660564s	178.886
4	UVM Events: Pr	efetch to GPU	J		C	.839494s	24.270
5	UVM Events: Pr	efetch to CPL	J		C	.863765s	11.855
6	UVM Events: Mi	grate to GPU			C).875621s	103.567

— 🗆 ts.managed.lego-cg1-qs-91.DriverVersion.550.54.14.20240408130441.nsys-rep ΞQ 📑 1x (i) <u>16 messages</u> +940ms +950ms +960ms -930ms +970ms the static table of the static static statics of head the shiften have the stand of a stand to be a stand of the stand egins: 0.920394s ids: 0.920398s (+3.615 µs) .7677 GiB/s ual address: 0xfffae50400 Migrate to GPU [103.567 ms] daDeviceSynchronize - F -Name - Description: TID Category 1798 UVM Events: Migrate to GPU 5 ms Begins: 0.875621s Ends: 0.979189s (+103.567 ms) 1798 ms Thread: 1798 1798 ms 1798 \Box

NVIDIA Nsight Systems 2024.5	.1		
<u>File View Tools Help</u>			
memloc_test.um_events.manage	d.accesscou	nters.lego-cg	1-qs
■ Timeline View ▼	© Options		
	0s •		15
▶ CPU (72)		100% 0	
- CUDA HW (0009:01:00.0 - NV	IDIA GH200	Kernel Memory	
▶ 87.9% Context 1			
 12.1% Unified memory 			
GPU Page Faults			
 100.0% Memory 			
86.6% HtoD transfer			
13.4% DtoH transfer			
 Threads (8) 			
▼ ✓ [1920] memloc_test •		0 to 100%	1 1
NVTX			C
CUDA API			Ì
Profiler overhead	2		
▼ ✓ [1934] memloc_test •		0 to 100%	
Profiler overhead			
6 threads hidden	-+	0 to 100%	
			4
Events View 🔻			
# Aname			
3 UV	/M Events: G	PU first touch	١
4 UV	/M Events: P	refetch to GP	U
5	/M Events: P	refetch to CP	U
6 UV	/M Events: N	ligrate to GPL	J
4			

Access Counter Based Migration

cess Cour	nter Ba	ased N	Лigra	tion	Che	CK Pr-	
s-91.DriverVersion.550.54.14.202404	08130441.nsys-rep >	< memloc_test.um	n_events.manage	ed.lego-cg1-qs-91	.DriverVersion.5	50.54.14.2024040813	Filing O
+870ms +880ms	s +890m	894.24ms +900m	s +91	Oms +	920ms	+930ms	16 messages +940ms
			Total and a				
	Write @ 0xfffc20	020000		Write @ 0xfffc2	0280000	Read @ 0xfffc	503000
	ແມ່ນເປັນໃນບັນເລ.ສາມາດໂອກດ໌.ສາດາານັບແລ	Regins: 0.894	143c	la.http:///www.co	annan an dhùthannan ann an an ann an ann ann ann ann	na 📕	Concession.
		Ends: 0.89414 HtoD transfer	48s (+5.503 µs) 655,360 bytes	ters			
		Throughput: 1 Virtual addres	10.913 GiB/s SS: 0xfffc2ca000				
cudaDeviceSy	cudaDevic	eSynchronize		cudaDeviceSyr	chronize	Cu CudaDeviceSyr	chronize
	1 1	11	İ	İ	11	11	
							▼
				Na	me 👻	inn	
	0 710785c	Duration	1020	Category		onte: Migrate to CDU	
	0.824302e	25 339 ms	1920		Begins:	0.861992s	、 III
	0.8496435	12.347 ms	1920		Ends: 0. Thread:	942108s (+80.117 ms) 1920)
	0.861992s	80.117 ms	1920				

- F

Q

Approaches for multi-process tools

- - Bugs in parallel programs are often serial bugs in disguise
- - ENV_VAR should be one set by the process launcher, unique ID
- Other tools: Use a launcher script, for late evaluation
- - --output=my_report.%q{SLURM_PROCID}
 - ./jacobi -niter 10

https://www.open-mpi.org/faq/?category=running#mpi-environmental-variables http://mvapich.cse.ohio-state.edu/static/media/mvapich/mvapich2-2.2-userguide.html#x1-32100013 https://slurm.schedmd.com/srun.html#SECTION_OUTPUT-ENVIRONMENT-VARIABLES

Tools usually run on a single process – adapt for highly distributed applications?

• Common MPI paradigm: Workload distributed; bug classes/performance similar for all processes • Not: Load imbalance, parallel race conditions; require parallel tools

 Ergo: Run tool N times in parallel, have N output files, only look at 1 (or 2, ...) • %q{ENV_VAR} supported by all the NVIDIA tools discussed here, embed environment variable in file name

• Evaluated only once tool starts running (on compute node) – not when launching job

srun nsys profile --trace=cuda,nvtx,mpi --force-overwrite=true

```
OpenMPI:
OMPI_COMM_WORLD_RANK
OMPI_COMM_WORLD_LOCAL_RANK
MVAPICH2:
MV2_COMM_WORLD_RANK
MV2_COMM_WORLD_LOCAL_RANK
Slurm:
SLURM_PROCID
SLURM_LOCALID
```


Using Multiple Reports in Nsight Systems

NVIDIA Nsight Systems 2022.4.1
<u>File View Tools H</u> elp
Untitled 8 * ×
= Timeline View -
CUDA HW (0000:03:00.0 - NVIDIA A100-5
39.0% Kernels
61.0% Memory
 Threads (9)
▼ 🗸 [31252] MPI Rank 0 👻
MPI
NVTX
CUDA API
Profiler overhead
✓ [31329] async 👻
7 threads hidden
[31255] ./jacobi
Events View 👻

Right-click a timeline row and select "Show in Events View" to see events here

				—		\times
	≣ର୍ (🗍 1x 🛛 🛆 🗕	warning	<u>is, 28 mes</u>	sages
	2,	5s		3s		
ri 1 🖞 📖						
						_
•						
						-11
Ψ "		TI P	· 1		1./1	
						-
		cudaF	cudaF			
		1-1 6	1			
L				9		
						-
						F
		Name	•			9
		Descript	ion:			
		1				_

Multi-report analysis recipes

Preview Feature

- Installation guide: <u>https://docs.nvidia.com/nsight-</u> systems/InstallationGuide/index.html#installingmulti-report-analysis-system
- Bundled installer script install.py, put dependencies into Python venv
 - Can be done by users themselves but could also be provisioned
- Run one of the recipes (or customize), example:
 - nsys recipe mpi_gpu_time_util_map --input *.nsys-rep
- Load generated .ipynb (embedded in nsys GUI)

- Nsight Systems by default collects stack samples
 - --sample=none to disable
- Display process/thread in events view or use tooltips

CPU Profiling Analyzing host-side call trees

= Timeline Viev	v •			Ξ Q	[¬] 1x <u>A 24 warnings, 12 messages</u>
	4s − m	ns +580ms + 4s 59	2,4ms +600ms +610ms	+620ms +630ms	+640ms +650ms
 Processes (218) 	3)				
▼ ● [306923]	./UnifiedMemoryPer				
CUDA HW	/ (000c:01:00.0 - NVIDI				
▶ 100.0%	Context 1				
▶ 0.0% Ur	nified memory				
 Threads (9))				
✓ ✓ [3069 CUDA Profile	923] UnifiedMemory -	cudaM cudaMe	Sampling point	cudaStreamSynch	ronize
Events View			libc-2.31.so!0x4001df785120 libcuda.so.525.60.13[8 Frames] UnifiedMemoryPerf!cudart196		
Livento view			UnifiedMemoryPerf!cudart332 UnifiedMemoryPerf!cudaMemcpyAsync UnifiedMemoryPerf!runMatrixMultiplyKernel() Name 🔻	
#	Name		UnifiedMemoryPerf!main	(<i>)</i>	Description:
63	libc-2.31.so!random		UnifiedMemoryPerf!Unresolved (_start?)		Call to cudaMemcpyAsync
64	libc-2.31.so!0x4001d	Jf80cfb4			Memory copies
65	UnifiedMemoryPerf!	fillMatrixWithRandomValues()			Begins: 4,58005s
66	cudaMemcpyAsync				Return value: 0
68	cudaMemcpyAsync				GPU: 000c:01:00.0 -
69	libc-2.31.so!0x40	01df785120			NVIDIA A100-PCIE-40GB
70	matrixMultiplyKerne	, 			Stream: /
72	cudaMemcpyAsync				Correlation ID: 7107
78	cudaStreamSynchror	nize			
79	matrixMultiplyKerne	, 			Call stack:
80	cudaMemcpyAsync				libcuda.so.525.60.13[7 Frames]
87	cudaStreamSynchror	nize			UnifiedMemoryPerf!cudart196 UnifiedMemoryPerf!cudart332
88	cudaFree				UnifiedMemoryPerf!cudaMemcpyAsync
90	cudaFree				runMatrixMultiplyKernel()
91	cudaFree				matrixMultiplyPerfRunner()
•					UnifiedMemoryPerf!main

- Example: Backtrace in CUDA sample
- Call stack leading up to cudaMemcpyAsync
- Periodic sampling: low overhead, but may miss some call chains
 - -cudabacktrace ensures CUDA API calls show backtrace, potentially higher overhead

CPU Profiling Zooming in

daMe Sampling point	cudaStreamSynchronize
Call stack at 4,59212s: libc-2.31.so!0x4001df785120 libcuda.so.525.60.13[8 Frames] UnifiedMemoryPerf!_cudart196 UnifiedMemoryPerf!cudaMemcpyAsync UnifiedMemoryPerf!runMatrixMultiplyKernel() UnifiedMemoryPerf!matrixMultiplyPerfRunner()	Name Description:
es()	Call to cudaMemcpyAsync■ Memory copiesBegins: 4,58005sEnds: 4,58108s (+1,027 ms)Return value: 0GPU: 000c:01:00.0 -NVIDIA A100-PCIE-40GBStream: 7Latency: 83,040 μs →Correlation ID: 7107
	Call stack: Ibc-2.31.so!0x4001df785120 Ibcuda.so.525.60.13[7 Frames] UnifiedMemoryPerf!_cudart196 UnifiedMemoryPerf!_cudart332 UnifiedMemoryPerf!cudaMemcpyAsync UnifiedMemoryPerf! runMatrixMultiplyKernel() UnifiedMemoryPerf! matrixMultiplyPerfRunner() UnifiedMemoryPerf!

CPU bottom-up (or top-down) profiling based on samples

• Linux perf events, e.g. cache misses

CPU Profiling Analyzing host-side call trees

• <u>https://docs.nvidia.com/nsight-systems/UserGuide/index.html#cpu-linuxperf</u>

Process [306923] ./UnifiedMemoryPerf (5.8%, 9 of 9 threads)

	Self, % 🔻	Module Name
igned int)	31 <mark>,52</mark>	/storage/code/cuda-samples/Samples/UnifiedMem
ed int, int, un	7,85	/storage/code/cuda-samples/Samples/UnifiedMem
ol, bool, bool,	7,85	/storage/code/cuda-samples/Samples/UnifiedMem
	7,85	/storage/code/cuda-samples/Samples/UnifiedMem
	8,58	/usr/lib/aarch64-linux-gnu/libcuda.so.525.60.13

•	0 s	 2s			 4s		
							-
	مط الفريمان ا	 يماديا ألب	l alla i	11.11	بالألاح	الدادية.	աստե

- cpu-socket-events=help)
 - Lists available metrics (events) on the CLI
- - hardware-performance-counters

CPU Core Metrics Profiling on Grace

• nsys profile --cpu-core-metrics=help (--cpu-core-events=help) --cpu-socket-metrics=help (--

 Generally recommended read: <u>https://docs.nvidia.com/grace-perf-tuning-guide/index.html</u> • In particular, see https://docs.nvidia.com/grace-perf-tuning-guide/index.html#measuring-workload-performance-with-

Measuring Workload Performance with Hardware **Performance Counters**

Many software performance analysis tools rely on event counts from hardware performance monitoring units (PMUs) to characterize workload performance. This chapter provides information about how data from PMUs can be gathered and combined to form metrics for performance optimization. For simplicity, the Linux perf tool is used, but the same metrics can be used in any tool that gathers hardware performance events from PMUs, for example, NVIDIA NSIGHT Systems.

CPU Socket Metrics --cpu-socket-events=61,71,265,273

		—		×
Ξ Θ 1χ	∧ 18 v	varnings 1	5 mess	ages
	~~~~	rannigo, i	0 111000	
+360ms +380ms	+400ms	+42	0ms	<u> </u>
		<u>\</u>		
				- 11
				4
				•
				1
Test [1.869 s]				
ata on GPU [139.894 ms]				
CPU [128.188 ms]				
		_		
				•
				-
				• I
				الم

NVIDIA Nsight Systems 2024.5.1			- 0
e <u>V</u> iew <u>T</u> ools <u>H</u> elp			
emloc_test.nsys-rep ×			
Ξ Timeline View     ▼	IS	🗉 Q 🗀 📩 1x	\land 18 warnings, 15 messag
2s	s +100ms +120ms	+140ms +160ms	+180ms
CPU (72)	% 0 <u></u>		
<ul> <li>Uncore</li> </ul>			
Socket_0_C2C0/rd_bytes_local			
Socket_0_C2C0/wr_bytes_local			
Socket_0_SCF/gmem_rd_data			
Socket_0_SCF/gmem_wr_total_byte			
<ul> <li>Processes (192)</li> </ul>			
	%		
<ul> <li>CUDA HW (0009:01:00.0 - NVIDIA</li> </ul>	el	ila irila irilati	
<ul> <li>100.0% Kernels</li> </ul>	vecadd_kernel(do	ouble *, const double *, const double *, int	t)
100.0% vecadd_kernel	vecadd_kernel(de	ouble *, const double *, const double *, int	t)
		Test [1.357 s]	
NVTX		Data on CPU [111.503 ms]	
		GPU [111.503 ms]	
<ul> <li>Threads (43)</li> </ul>			
🗢 🔽 [11675] memloc_test 💂	%		
		Teet [1.060 e]	
NIV TY		rest [1.009 s]	
INVIA		CDIL[111_602_mol	
		GPU [111.095 ms]	
CUDA API		cudaDeviceSynchronize	
Profiler overhead	a		
✓ [11712] memloc_test +	%		
	4		Þ

![](_page_26_Picture_6.jpeg)

![](_page_27_Picture_0.jpeg)

![](_page_27_Picture_1.jpeg)