

Nsight Systems - Introduction Robert Dietrich - February 8, 2022

Typical Optimization Workflow



Iterate until desired performance is achieved

Legacy Transition



System-Wide Application Tuning

Maximize your GPU Investment

Locate optimization opportunities

- Visualize millions of events on a timeline
- See gaps of unused CPU and GPU time

Balance your workload across multiple CPUs and GPUs

- CPU utilization and thread state
- GPU streams, kernels, memory transfers, etc.

Multi-platform support

- Linux, Windows and Mac OS X (host-only)
- x86-64, Power9, ARM server, Tegra (Linux & QNX)





File <u>View</u> <u>Tools</u> <u>H</u>elp



Command Line Interface (CLI) Statistics and Export to SQLite, JSON, etc.

Command Line Interface

The Nsight Systems CLI provides several different commands

- Basic profiling session nsys profile ./app
- Interactive sessions (scriptable) nsys start|launch|stop|cancel nsys session list nsys status|shutdown
- Statistics and export nsys stats | export (export to sqlite, hdf, text, json, info)



CLI Profiling - Some Useful Switches

API tracing

```
-t, --trace=cuda,nvtx,osrt,opengl
```

(cublas, cusparse, cudnn, mpi, oshmem, ucx, openacc, openmp, vulkan, none)

Overwrite existing report

```
-f, --force-overwrite=[true|false]
```

Summary statistics (profile output on command line)

```
--stats=[true|false]
```

Report file name

```
-o, --output=report#
```

```
(patterns for hostname, PID and environment variables)
```

CLI Profiling - Some Useful Switches

Callstack sampling

- -s, --sample=[**cpu**|none]
- --sampling-period=number of CPU Instructions Retired events
- -b, --backtrace=[**lbr**|fp|dwarf|none]
- --samples-per-backtrace={1..12}
 - (The number of CPU IP samples collected for every CPU IP sample backtrace collected.)

Set the paranoid level: "sudo sh -c 'echo 1 >/proc/sys/kernel/perf event paranoid'

CUDA memory usage

```
--cuda-memory-usage=[true |false]
```

```
(Use nsys profile --help for a list of available options.)
```

CLI Profiling - MPI Programs

Single Node

nsys profile [nsys_args] mpirun [mpirun_args] your_executable

⇒ This creates one report file.

Multiple Nodes

mpirun [mpirun args] nsys profile [nsys args] your_executable

Set output report name with -o report_name_%q{OMPI_COMM_WORLD_RANK} (for OpenMPI, PMI_RANK for MPICH and SLURM_PROCID for Slurm)

CLI Profiling - Additional Output

WARNING: The command line includes a target application therefore the CPU context-switch scope has been set to process-tree. Collecting data...

... APP OUTPUT ...

Temporary data is written to **/tmp/nvidia/nsight_systems** by default. Set **TMPDIR** to specify another location.

Processing events...

Saving temporary "/tmp/nsys-report-2b96-9038-d0bd-2600.qdstrm" file to disk...

CLI Stats Output

- CUDA API, kernels and memory operations (by time and by size)
- OS runtime

CUDA

Time

		NVTX P	NVTX Push-Pop Range Statistics:										
NVTX		Time(%) Total Ti	ne (ns) I	nstances	Average	Minimum	Maximum	Range				
		80 18	,0 16.193. ,0 3.680.9	376.534 937.672	21.268 8.072	761.396,0 456.013,0	6.823 1.726	2.438.848 1.932.681	MPI:MPI_Waitall MPI:MPI_Allreduce				
		0	,0 88.5	587.708 388.066	4 21,268	22.146.927,0	21.714.183	22.579.671 297.466	MPI:MPI_Init MPI:MPI Isend				
		0	,0 24.	282.234	21.268	1.141,0	872	16.931	MPI:MPI_Irecv				
		0	,0 8. ,0 3.	793.048 314.158	4 28	2.198.262,0 136.219,0	2.181.292 2.302	2.215.232	MPI:MPI_Finalize MPI:MPI_Barrier				
Kern	el Statistics:	0	,0	266.630	32	8.332,0	832	104.980	MPI:MPI_Reduce				
(%)	Total Time (ns)	Instances	Average	Minimum	Maxim	um			Name				
7,0	7.344.350.347	8.568	857.183,0	851.54	5 876.	922 device_te	a_leaf_ppcg_	solve_calc_s	d_new(kernel_info_t	, d			
6,0	7.229.217.310	8.568	843.746,0	839.32	899.	482 device_te	a_leaf_ppcg_	solve_update	_r(kernel_info_t, d	loub			
0,0	2.050.961.573	2.010	1.020.378,0	1.005.849	9 1.036.	569 device_te	a_leaf_cg_so	lve_calc_ur(kernel_info_t, doub	le,			
9,0	1.879.919.365	2.010	935.283,0	913.75	4 1.214.	808 device_te	a_leaf_cg_so	lve_calc_w(k	ernel_info_t, doubl	.e*,			
5,0	1.008.596.179	1.980	509.392,0	501.469	9 521.	756 device_te	a_leaf_cg_so	lve_calc_p(k	ernel_info_t, doubl	.e, (
0,0	61,606.046	30	2.053.534,0	1.176.600	0 2.890.	541 device_te	a_leaf_calc_	<pre>rrn(kernel_i</pre>	nfo_t, double const	*.			
0,0	38.094.843	12.336	3.088,0	1.88	7 13.	184 void redu	ction <double< td=""><td>, (REDUCTION</td><td>_TYPE)0>(int, doubl</td><td>e*)</td></double<>	, (REDUCTION	_TYPE)0>(int, doubl	e*)			

Launch Application via GUI

•		NVIDIA Nsight Systems 2021.3.1		~ ^ (
<u>F</u> ile <u>V</u> iew <u>T</u> ools <u>H</u> elp				
Project Explorer SimpleCUBLAS MaiTastSuite	Localhost connection	✓ Target is ready More info		
mpriestsuite	SSH connections (3)			
	Solution Solution	0 events tions Retired' events counted before a CPU instruction p ected. The smaller the sample period, the higher the san icantly increase the size of result file(s).	Start profiling manually Start profiling after 10,0 \$ seconds Start profiling after 100 \$ frames Limit profiling to 10,0 \$ seconds	
	Iarget application Mode: Specify process launch options below Command line with arguments: mpirun -np 1 ./mpi_test_suite Working directory: /home/rdietrich/testing/mpi/mpi-test-suite		Edit arguments	 Limit profiling to 600 ♀ frames Hotkey Start/Stop (not available in console apps)
	Environment variables Trace fork before exec	File View Tools Help	A	
	Collect OS runtime libraries trace Collect CUDA trace Collect OpenMP trace	<u>Open</u> <u>A</u> dd Report (beta) <u>I</u> mport <u>Export</u>	Ctrl+N MpiTe Ctrl+O Ctrl+T Sele Ctrl+I Last Ctrl+E Selec	ect target for profiling Constrained by the set of
	Collect GPU context switch trace	<u>C</u> lose MpiTestSuite <u> </u>	Ctrl+W Ctrl+Q	

Eastura Salaction	► ✓ Collect CPU context switch trace					
l'eature selection	۲	Collect OS runtime libraries trace				
 ✓ Sample target process 	•	✓ Collect CUDA trace				
Sampling Period: 1,000,000 events The sampling period is the number of 'CPU Instructions Retired' events counted bef collected. If configured, call stacks may also be collected. The smaller the sample periods will increase overhead and significantly increase the size of result Collect call stacks of executing threads Backtracing algorithm Current settings: use DWARF debug information Symbol locations No directories with symbol files. When stripped libraries (e.g. *.so files) are used on the target, specify here director symbols resolved. For best backtraces, specify the following compiler flags: on x86_64: -g Note that stripped binaries typically do not contain the debug information. Conside	•	 Flush data periodically 10,00 \$ seconds Skip some API calls Collect GPU memory usage Collect UM CPU page faults Collect UM GPU page faults Collect cuDNN trace Collect CuDA backtraces Collect CUDA backtraces Collect OpenMP trace 				
Hote diacatipped binaries typically do not contain the debug information. Consid	•	Collect GPU context switch trace				
Target application	•	Collect GPU metrics				
Mode: Specify process launch options below Command line with arguments:	Collect NVENC trace					
mpirun -n 2 /home/rdietrich/testing/mpi/hello_mpi/hello_mpi	-	V Collect NVTX trace				
Working directory:	•	Collect OpenGL trace				
/home/rdietrich/testing/mpi/hello_mpi	•	Collect Vulkan trace				
		Communication profiling options (MPI, SHMEM, UCX)				

Feature Highlights



File <u>View</u> <u>Tools</u> <u>H</u>elp



CPU Cores Workload

See CPU core utilization by application's threads Locate idle time on CPU cores



CPU Thread Activity



🕺 NVIDIA

OS Runtime Libraries

	3s +437.9ms +437.95ms +438ms + 3s 438.	0614ms +438.1ms +438.15ms +438.2ms	+438.25ms +438.3ms +438.35ms +438.4ms +438.45ms +438.55ms +				
✓ 🗹 [12363] lm →							
OS runtime libraries	pthread						
NVTX		MPD_TIM	STEP [760.124 µs]				
0.033528		MPD_SYNCHRONIZE [548.501 µs]					
CUDA API			cudaStreamSynchronize				
✓ 🗹 [12364] lm →							
OS runtime libraries	pthread_mutex_lock	Waiting	-				
NVTX		Duration: 6.818 μs	ESTEP [766.412 µs]				
(**) (10)		Call stack at 3.438s:	MPD_SYNCHRONIZE [471.450 µs]				
CUDA API		libpthread-2.23.so!_pthread_mutex_lock	cudaStreamSynchronize				
✓ 🗹 [12366] lm →		libcuda.so.384.81!0x7fa6033595f6 libcuda.so.384.81!0x7fa603359628 libcuda.so.384.81!0x7fa60326b825	1				
OS runtime libraries	p) pthread_mutex_lock	libcuda.so.384.81!0x7fa60326c958 libcuda.so.384.81!cuMemcpyAsync libcudart.so.9.0.176!0x7fa619dfddfd					
NVTX		libcudart.so.9.0.176!0x7fa619dda573	IMESTEP [780.810 µs]				
		Im!ZDivMultiGPUMapper:schedule_send()	MPD_SYNCHRONIZE [483.652 µs]				
CUDA API	cudaMe	Im!Im:rdme::MGPUMpdRdmeSolver:run_next_timestep()	cudaStreamSynchronize				
✓ 🗹 [12357] lm →		libpthread-2.23.so!start_thread libc-2.23.so!_clone					

Includes backtraces of long running functions

CUDA API

- Trace CUDA API Calls on OS thread
- See when kernels are dispatched
- See when memory operations are initiated
- Locate the corresponding CUDA workload on GPU



GPU Utilization



GPU Utilization



... reveals gaps where there were valleys.

CPU-GPU Correlation



CPU-GPU Correlation & Location Assistance



CUDA Memory Transfers



cudaMemcpyAsync behaving synchronous Device to host pageable memory Mitigate with pinned memory

Project 5 x Report 11 x 🖾 Q 1x 🗌 Timeline View w +897ms +897.5ms +898ms 1s 898.508ms +899ms +899.5ms +900ms 1s -CPU (12) Threads (8) ✓ [22032] UvmVectorAdd mmap sem timedwait OS runtime libraries CUDA API cudaDeviceSynchronize cudaFree CUDA profiling data flush overhead Profiler overhead ▼ ✓ [22038] UvmVectorAdd - 337 px; 0 msec OS runtime libraries sem timedwait poll 6 threads hidden... -+ 1337 px: 0 msec - CUDA HW (0000:02:00.0 - Gef . 85.7% Context 1 ▼ 14.3% Unified memory UVM GPU page fault @ 0x7f4fa4000. UVM GPU page fault @ ... UVM, **GPU Page Faults** 100.0% Memory UVM GPU page fault Begins: 1.89844s 69.8% HtoD transfer Ends: 1.89856s (+118.112 us) Virtual address: 0x7f4fa4080000 30.2% DtoH transfer # of page fault groups: 5 CUDA UVM CPU Page Faults Memory access type: Write 1.074 4

Events View -

					Name 👻
¥	* Name	Start	Duration	GPU	Description:
1	UVM GPU page fault @ 0x7f4fa4000000	1.89747s	564.124 µs	GPU 0	UVM GPU page fault
	UVM GPU page fault @ 0x7f4fa4010000	1.89804s	397.469 µs	GPU 0	Begins: 1.89747s
	UVM GPU page fault @ 0x7f4fa4080000	1.89844s	118.112 µs	GPU 0	Virtual address: 0x7f4fa4000000
					# of page fault groups: 1 Memory access type: Read

CUDA Unified Memory CPU & GPU Page Fault Trace

💿 nvidia

CUDA Libraries & DL Frameworks

cuDNN, cuBLAS, cuSPARSE & TensorRT, Tensorflow

🗹 [6307] sample 👻								
OS runtime libraries	pthread_mute)			-				fflush
CUDA API	gemmSN_NN_kernel_halt	add	tens	nch cudaEvent	R	softmax_fw_kernel	 	cudaEventSync
cuDNN		(in 1997)	AddTensor			SoftmaxForward		
cuBLAS	Hgemm							
💌 🗹 [176211] conjugateGradi 🕶								
cuSPARSE	1179 px; 0 n	cusparseScsrsv2_solve [24,662 µs	;]	cusparseScsrsv2_solve [14	,731 µs]		cusparseSpMV [17,668 µs]
CUDA API	1179 px; 0 msec	cudaMemset	csrsv2_solve_lo	cudaMe cudaMe	rsv2_sol	copy_kernel	binary_sea	rch load_balan
Profiler overhead	1179 px; 0 msec							
7 threads hidden – +	1179 px; 0 msec							2
CUDA HW (0000:65:00.0 - NVIDI/	1179 px; 0 msec		_					
▶ 99.1 <mark>%</mark> Kernels	1179 px; 0 msec		V	oid csrsv2_solve_lower_nontrans_	byLevel_kernel <float< th=""><th>, (int)5, (int)3>(int, int, cons</th><th>t T1 *, const int *, cons</th><th>st int *, const T1 *, T1 *, i</th></float<>	, (int)5, (int)3>(int, int, cons	t T1 *, const int *, cons	st int *, const T1 *, T1 *, i
 0.9% Memory 	1179 px; 0 msec							
8.091NVTX (cuSPARSE)	1179 px; 0 msec				cusparseScsrsv2_so	olve [1,285 ms]		

Host operation is mapped to GPU operations

User annotations APIs for CPU & GPU

NVTX, OpenGL, Vulkan, and Direct3D performance markers



Example: Visual Molecular Dynamics (VMD) algorithms visualized with NVTX on CPU

Additional Views

Statistical Sampling

IP sample ← Code path 1 ←

Code path 2 leading to -IP sample

Bottom-Up View V Process [9695] vmd_LINUXAMD64.11 (3 of 19 threads)

Filter... 99.82% (23,260 samples) of data is shown due to applied filters.

Symbol Name	Self, %	Module Name
✓ VolumetricData::compute_volume_gradient()	20.14	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
VolumetricData::compute_volume_gradient()	20.14	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ BaseMolecule::add_volume_data(char const*, double const*, double const*, double const*, double const*, int, int, int, float*)	18.30	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
VMDApp::molecule_add_volumetric(int, char const*, double const*, double const*, double const*, double const*, int, int, int, float*)	18.30	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ obj_segmentation(void*, Tcl_Interp*, int, Tcl_Obj* const*)	18.30	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
[Max depth]	18.3 <mark>0</mark>	[Max depth]
✓ BaseMolecule::add_volume_data(char const*, float const*, float const*, float const*, float const*, int, int, int, float*, float*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ MolFilePlugin::read_volumetric(Molecule*, int, int const*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
VMDApp::molecule_load(int, char const*, char const*, FileSpec const*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ text_cmd_mol(void*, Tcl_Interp*, int, char const**)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ TclInvokeStringCommand	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ TclEvalObjvInternal	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ TclExecuteByteCode	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ TclCompEvalObj	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ TclEvalObjEx	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ Tcl_RecordAndEvalObj	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ TclTextInterp::evalFile(char const*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ VMDApp::logfile_read(char const*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
✓ VMDreadStartup(VMDApp*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
[Max depth]	1.84	[Max depth]
> 0x7f10ca7022d6	5.13	/usr/lib64/libcuda.so.390.25
> obj_segmentation(void*, Tcl_Interp*, int, Tcl_Obj* const*)	3.44	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11

shows statistics from periodic call-stack backtraces

Statistical Sampling - Filtering

	2.4s	2.65	2.85	35	3.2s	3.4s	3.6s	3.85	
CPU (8)									
 Threads (19) 									
▼ 🗹 [23600] vmd_LINU -	<u></u>	and the second	ing and di						
OS runtime libraries	1	ninadadada mi a din							
						VME) process user script	(s) [2.506 s]	
	WMDreadSt		drc [552.183	Les Cartes			VMDre	adStartup()	
	The Charles of Charles							Segmentat	
NVTX ([Default])				MDApp			Segmentation:	:segment()	
						Watershee	::watershed()[653.	451 ms]	
						(Watershed::	watershed_gpu() [63	19.827 ms]	
CUDA API									
	•								
Bottom-Up View 👻 Process [2	23600] vmd_L	INUXAMD64 (3 of 1	9 threads)						
Tilter 9.68% (2,349 sar	mples) of data	is shown due to a	pplied filters	Time filter: 2	.94 to 3.22 (0.2	28 seconds or 2.8	%).		
Symbol Name		Self. %	▲ Module M	lame					
 IO vfscanf 		21	.97 /usr/lib64	4/libc-2.17.so	N. Contraction of the second s				
▼ IO vfscanf		21	.97 /usr/lib64	4/libc-2.17.so					
 isoc99 vsscanf 		21	.97 /usr/lib64	4/libc-2.17.so					
 _isoc99_sscanf 	21	.97 /usr/lib64	4/libc-2.17.so						
atoifw	20	.65 /home/jo	hns/vmd/LINU	JXAMD64/vmd	LINUXAMD64				
read pdb structure	0	72 /home/jo	2 /home/johns/vmd/LINUXAMD64/vmd LINUXAMD64						
get psf atom		0	.60 /home/jo	0 /home/johns/vmd/LINUXAMD64/vmd LINUXAMD64					
VolumetricData::compute_vo	lume gradient	18	26 /home/johns/vmd/LINUXAMD64/vmd_LINUXAMD64						
read ccp4 data(void*, int, flo	8	.77 /home/jo	7 /home/johns/vmd/LINUXAMD64/vmd_LINUXAMD64						

Events View

Events View

*

					Name	-
#	▼ Name	Start	Duration	TID	*	Description:
149153	device_tea_leaf_ppcg_solve_calc_sd_new	11,4088s	3,265 µs	390019		Call to cudaMemcpy
149154	device_pack_bottom_buffer	11,4088s	3,247 µs	390019		Memory copies
149155	🕐 cudaMemcpy	11,4088s	1,838 ms	390019		Begins: 11,4088s
149174	MPI_Isend	11,4106s	1,564 µs	390019		Return value: 0
149175	MPI_Irecv	11,4106s	1,420 µs	390019		Correlation ID: 40146
149176	MPI_Waitall	11,4106s	1,772 ms	390019		Call stands
149199	cudaMemcpy	11,4124s	77,788 µs	390019		Call Stack: libcuda.so.460.27.04[12 Frames]
149201	device_unpack_bottom_buffer	11,41255	5,232 µs	390019		libcuda.so.460.27.04!cuMemcpyDtoH_v2
149202	device_tea_leaf_ppcg_solve_update_r	11,4125s	3,334 µs	390019		libcudart.so.11.1.74[Z Frames] libcudart.so.11.1.74[cudaMemcpy
149203	device_tea_leaf_ppcg_solve_calc_sd_new	11,4125s	3,268 µs	390019		tea_leaf!
149204	device_pack_bottom_buffer	11,4125s	3,031 µs	390019		tealeafCudaChunk::packUnpackAllButters() tea leaflcuda pack buffers
149205	cudaMemcpy	11,4125s	1,853 ms	393019		tea_leafltea_module_MOD_tea_exchange
149224	MPI_Isend	11,4143s	2,081 µs	390019		update halo module MOD update halo
149225	MPI_Irecv	11,4143s	1,191 µs	390019		tea_leafl
149226	- MPI_Waitall	11,4143s	1,786 ms	390019		tea_leat_ppcg_module_MOD_tea_leat_run_p pcq_inner_steps
149227	libucp.so.0.0.0!ucp_worker_progress	11,4144s	-	390019		tea_leaf!_tea_leaf_module_MOD_tea_leaf
149228	libucp.so.0.0.0!ucp_worker_progress	11,4144s	1.00	390019		[Unknown]!0x0
149229	libopen-pal.so.40.20.5!opal_progress	11,4145s	12	390019		[Broken backtraces][[Broken backtraces]
149230	libuct.so.0.0.0luct_mm_iface_progress	11,4146s	-	390019		
149231	libonen-nal so 40 20 51onal progress	11 4147s	_	390019	*	

Expert System

									CUDA Async	Memcpy with P	ageable Mem	0
- File <u>V</u> iew <u>T</u> ools Help									CUDA Synch	ronous Memcpy	/	
profile_circe-n011_506451_0.qdrep.	qdrep 🗙								CUDA Synch	ronous Memset		
■ Timeline View ▼								📼 Q 1x	CUDA Synch	ronization APIs		
55.5		+676,05ms	+676,1ms	+676,15	ns +6	76,2ms	+676,25ms	+676,3ms	CUDA GPU S	tarvation		
CUDA API 📮	cudaStreamS	ec elemen	elem	el)	udaMemc	cudaMemcpyAsyn	c cuda)		CUDA GPU L	ow Utilization		
OS runtime libraries						pthread_cond_wait		•	VULKAN GPI	U Starvation		
10 threads hidden — 🕂	1179 px; 0 m	sec							VULKANGR			
- CUDA HW (0000:34:00.0 - Te	1179 px 0 m		1 L		4				VOLKAN GP	o Low Othization	1	
 [All Streams]	ncciAll m	sec			1 I		- i					
▶ 60.8% Default stream 1	1179 px; m	sec		i i	Y I		- i					
10.79 39.2% Stream 29	ncclAll 0 m	sec										
	4	coc.									Þ	-
Expert System View												
Expert System view											8 Setting	15
CUDA Async Memory with Pageah	ale Memory 💌	Duration + St	art	Src Kind	Dst Kind	Bytes	PID	Device ID	Context ID	Stream ID	API Name	
coord of the mentery man agead	a memory	2,048 µs	6,38792s	Device	Pageable	8 B	75475	0	1		7 cudaMemcp	
The following APIs use PAGEABLE which causes asynchronous CUDA	memory — A memcov	2,048 µs	6,8334s	Device	Pageable	4 B	75475	0	1	1	7 cudaMemcp	
operations to block and be execut	ed	2,016 µs	2,5394s	Device	Pageable	4 B	75475	0	1		7 cudaMemcp	
synchronously. This leads to low G utilization.	PU	2.016 µs	3.90617s	Device	Pageable	48 B	75475	0	1		7 cudaMemcp	
		2.016 µs	4.25257s	Device	Pageable	4 B	75475	0	1	8	7 cudaMemcp	
suggestion: If applicable, use PIN memory instead.	NED	2,016 µs	5,67617s	Device	Pageable	48 B	75475	0	1		7 cudaMemcp	
CI I command:		2,016 µs	5,9572s	Device	Pageable	8 B	75475	0			7 cudaMemcp	
nsys analyze -r cuda-async-memcp	y /mnt/data/	2,016 µs	5,97088s	Device	Pageable	4 B	75475	0	1		7 cudaMemcp	
traces/qdrep/nccl/profile_circe- n011_506451_0_solite		4			· · · ·				1	1		*
non_soursi_o.squee												-

GPU Metrics Sampling

	• 16.6s 16.7s 16.8s	16.9s 17s	17.1s 17.2s 17.3s 17.4s	17.5s 17.6s 17.7s 17.8s 17.9s					
 CPU (256) 									
 GPU (A100 Graphics Device - 0 		I REFERE AND DATA DATA DE							
▼ GPU Metrics	lative line la		• Is my GPU full? Sufficient grids size & streams?						
SYS Clock Frequency	and common and a second second second second	to management and the							
GR Active	aber any a little in the second s	and a second second second	 Is my instruction rate low (possibly IO bound)? 						
SM Active	line and the state	and a state of the							
SM Instructions			Am I using tensor cores	S?					
SM Warp Occupancy		andraiga sha maas	Can I see GPU Direct R	DMA/Storage or other transfers?					
DRAM Bandwidth	hi a ci a bi hi hi i	kierandina kuna	Custom wide CDU shee	an en til e e					
PCIe Bandwidth CPU (4100 Craphics Device 1)		•	System-wide GPU obse	rvation					
 GPU (A100 Graphics bevice - C GPU Metrics 	and the second s	bach therein the a function the	(no app required but sudo or	regkey					
GPC Clock Frequency			(no upp required, but sudo or						
SYS Clock Frequency			10kHz default can be i	ncreased depending on GPU					
GR Active		and and a second and a second		J					
 SM Active SM Instructions 	ulter Tree Iner		Metrics:						
	LEALANDER IN JUNE JUNE AND JUNE AND JUNE AND								
SM Warp Occupancy	Man Man And								
DRAM Bandwidth			SM utilizations	IO throughputs					
PCIe Bandwidth		tabilitatia Estas	- CMa activa						
 GPU (A100 Graphics Device - 0 	oc		• SIMS active	• PCIe					
GPU Metrics GPU (A100 Graphics Device - 0	or	a his bar a ball har and the start of	Instructions	 NVL ink 					
GPU Metrics	NAME AND ADDRESS OF AD	de lasta danstitu							
 GPU (A100 Graphics Device - 0 	oc		TensorCores						
GPU Metrics GPU (A100 Craphics Doutes - 0	reiter in the second sector is	al leased descend bet also be a							
GPU Metrics	The second secon	the design of the second second	Warp occupancy						
 GPU (A100 Graphics Device - 0 	oc		i i a poccapariej						
GPU Metrics GPU (A100 Creative Device 1)	to be he he have been been here here here here here here here h		(including unalloc	ated slots)					
GPD (A100 Graphics Device - C			(
[All Streams]			dgrad2d_grouped_direct_kernel	dgrad2d_grouped_direct_kernel dgrad2d_grouped_direct_kernel					
[29106] ovthon									
CUDA API		msetAsync	a/EventSyndrronize	autorentseard					
✓ [24832] python -				TensorFlow on 8xGA100 at 20kHz					
CUDA API		1 1 1 1 1 1 1 1 1 1	ouStreamSyndronize						

Communication Libraries & Multi-Node Support

OpenSHMEM, MPI and UCX

 [1430905] SHMEM PE 1 						
OS runtime libraries	971 px; 0 msec					
	shmem	shmem_float_p [7,145]	shme	shme	m_barrier_all [18,019 μs]	shmem_finalize [1,279 s]
SHMEM	971 px ; 0 msec		shmem	float p		shmem_barrier_all [7,3
Profiler overhead	971 px; 0 msec		Begins:	0,472616s		
 [1430917] 	971 px; 0 msec		Ends: 0, Thread:	472619s (+2,470 μs) 1430905		

👻 🔽 [174635] MPI Rank 0 🕞				
MPI	MPI_I:	send [7,319 µs]	MPI_Irecv [7,585 µs]	MPI_Waitall [11,493 µs]
 [174635] MPI Rank 0 + MPI UCX Start & End Profiler overhead 	ucp_tag_send_nbx UCP tr		ucp_tag_recv_nbx UCP transf	ucp_rkey
- Con	i o i o pago i i occ	ucp_tag_send_nbx	use the cond aby LUCD tennelse preserves (34.403) and
Start & End	1019 px; 0 mse	Ends: 0,177988s (+3,990 µs)	ucp_tag_send_nbx OCP transfer processing [21,163	_nbx UCP transfer processing [14,243 µs]
Profiler overhead	MPI_Isend [7,319 µs] MPI ucp_tag_send_nbx UCP tr ucp_tag_it 1019 px; 0 msec ucp_tag_send_nbx Begins: 0,177984s ucp_tag_it 1019 px; 0 msec interaction 1019 px; 0 msec interaction			
6 threads hidden 🗕 🕂	1019 px; 0 msec	Category: UCP transfer submit		

Completion tracking of non-blocking UCP communication operations

NVSHMEM and NCCL

■ Timeline View ▼				<u>.</u>	Q 1x	0		(1) 23 messages
15	+521,52ms +521,54ms +521,56ms	+521,58ms +521 1s 521,	6087ms1,62ms +521,64ms +521,66ms	+521,68ms	+521,7	ms	+521,72ms	+521,74ms
 [19621] NVSHMEM PE 0 	1063 px; 0 msec							
NCCL	1063 px; 0 msec	ncclAllReduce [26,3.	1					
	nvshmem_malloc [36,690 µs]	nvshmemx_int_sum_r.				0	nvshmem_free	2 [91,979 μs]
NVSHMEM	1063 px; 0 m(n)					nvshm	em_quiet [
CUDA API	cuSt ba cuStre accu	ncciKe].	. cor cudaStreamSynchronize	cud		CuS	treamSy b	arr cuStrea
Profiler overhead	1063 px; 0 msec							
▼ CUDA HW (0000:06:00.0 -	1 1063 px msec	L						
[All Streams]	1063 px msec bar a		cclKernel_AllReduce_RING_LL_S					barrie
79.5% Default stream 7	1063 px; 0 msec		ncclKernel AllReduce RING LL Sum int32 t					
7.335 ⁺ 19.8% Stream 22	1063 px		Begins: 1,52161s					barrie
			Ends: 1,52165s (+41,504 µs) arid: <<<1, 1, 1>>>					Þ
Events View 👻			block: <<<64, 1, 1>>>					
			Static Shared Memory: 41.472 bytes		Name	•		Q
# • Name			Registers Per Thread: 96	TID	-	Descripti	ion:	
730 💌 nvshme	emx_int_sum_reduce_on_stream		Local Memory Per Thread: 0 bytes	19621		nvshme	mx int sum r	reduce on stream
1 731 • ncc	AllReduce		Shared Memory executed: 98.304 bytes	19621		Begins: 1	,52158s	Euch
732 cu	daStreamIsCapturing		Shared Memory Bank Size: 4 B Launched from thread: 19621	19621		Thread:	19621	1040
733 cu	daStreamWaitEvent		Latency: ←10,563 µs	19621				
734 nc	clKernel_AllReduce_RING_LL_Sum_int32_t		Stream: Stream 20	19621	-			

OpenMP - Correlation Highlighting



OMPT-capable OpenMP runtime is required.

Loading Multiple Reports into one Timeline

<u>File View T</u> ools <u>H</u> elp			
New Project	Ctrl+N		
<u>O</u> pen	Ctrl+O	⊠ Θ1v □	(i) 17 messages
Add Report (beta)	Ctrl+T	2° 3° de 5e 6e 7e	85
Import	Ctrl+I		
Export jacobi_2mpi.qdrep [2 repo	orts] Ctrl+E		
<u>Close jacobi_2mpi.qdrep [2 repor</u>	ts] Ctrl+W		
⊛ E <u>x</u> it	Ctrl+Q		
		Jacobi solve [3,472 s] Jacobi solve [1,811 s]	
NVTX	1179 px; 0 msec		
CUDA API	1179 px; 0 m cuda cudaHos.	cudaHo	
Profiler overhead	117 k; 0 msec		
9 threads hidden – +	1179 px; 0 msec		
 CUDA HW (0000:06:00.0 - Tesla 	1179 px; 0 msec	· · · · · · · · · · · · · · · · · · ·	
 prm-dgx-16 (0:0) 	1179 px; 0 msec		
 CPU (80) 	1179 px; 0 msec		
▼ Threads (10)	1179 px; 0 msec		
👻 🔽 [19265] MPI Rank 0 👻			
OS runtime libraries	112 ioctl	iocti	
MPI	11 MPI_Initsec		j
NVTX	1179 px; 0 msec	Jacobi solve [3,474 s] Jacobi solve [1,811 s]	
2.841 CUDA API	1179 px; 0 m cuda cudaHo)cudaHo	

Nsight Systems - Summary

Today

- Mainly used for single server profiling runs
- Focused on GPU workloads, API trace, thread samples
- Cluster profiling
 - Load multiple reports into one timeline
 - MPI (including communication parameters), OpenSHMEM, UCX tracing
 - NIC metrics sampling (congestions, read/write bandwidth)

Roadmap

- Improve multi-node support, e.g. correlation across nodes, switch statistics, etc.
- Determine latency, congestion, hot devices (NICs, switches, network storage)

THANK YOU!

Download	https://developer.nvidia.com/nsight-systems NOTE: website version is newer than CUDA Toolkit version
Docs	https://docs.nvidia.com/nsight-systems/index.html
Forums	https://devtalk.nvidia.com
Email	nsight-systems@nvidia.com
Blogs	 <u>https://developer.nvidia.com/blog/nvidia-nsight-systems-containers-cloud</u> <u>https://developer.nvidia.com/blog/nsight-systems-exposes-gpu-optimization</u> <u>https://developer.nvidia.com/blog/understanding-the-visualization-of-overhead-and-latency-in-nsight-systems</u> <u>https://developer.nvidia.com/blog/nvidia-tools-extension-api-nvtx-annotation-tool-for-profiling-code-in-python-and-c-c/</u>

Hands-on TeaLeaf

Hands-on TeaLeaf

e View Tools Help Er50.qdrep x report.qdrep x profile_circe-n011_506451_0.qdrep.qdrep x laplace2d_baseline.qdrep x taplace2d_data.qdrep x tealeaf.qdrep x Timetine View Timetine View				NVIDIA Nsight Systems	2021.3.1			
Hersbacker x report.gdrep x profile_circe-n011_506451_0.qdrep.qdrep x laplace2d_baseline.qdrep x laplace2d_data.qdrep x tealeaf.qdrep x Immeraus (TT) Immeraus (e <u>V</u> iew <u>T</u> ools <u>H</u> elp							
Timeline View Image: 1x	-r50.qdrep 🗙 report.qdrep 🗙 p	profile_circe-n011_506451	1_0.qdrep.qdrep 🗙 laplace2	d_baseline.qdrep 🗙 laplace2	d_data.qdrep 🗙 tealeaf.qdre	р×		
Ba< + 4904ms +906ms +906ms +914ms +914ms +916ms +920ms +924ms IMIREAUS (TT) Imiteaus (TT) <th>➡ Timeline View 👻</th> <th></th> <th></th> <th></th> <th></th> <th>🖾 Q 1x 🗍</th> <th>A warnings, 2</th> <th>25 message</th>	➡ Timeline View 👻					🖾 Q 1x 🗍	A warnings, 2	25 message
• ♥ ♥ [52199] MPI Rank 1 • MPI MPI MVTX Update_halo [3,701 ms] update_halo [3,672 ms] update_halo [3,689 ms] update_halo [3,699 ms] update_halo	 Inreads (11) 	+904ms +906ms	+908ms +91	0ms +912ms	+914ms +916ms	+918ms +920ms	+922ms +924m	s i i i i
MPI MPI_Waitali [NVTX update_halo [3,672 ms] update_halo [3,657 ms] update_halo [3,695 ms] update_halo [3,698 ms] update_halo [3,691 ms] <td>👻 🗹 [52199] MPI Rank 1 👻</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	👻 🗹 [52199] MPI Rank 1 👻							
NVTX update_halo [3,701 ms] update_halo [3,672 ms] update_halo [3,657 ms] update_halo [3,668 ms] update_halo [3,669	MPI				MPI_Waitall [)		
CUDA API cudaMemcpy cudaMemcpy <thcudamemcpy< th=""> cudaMemcpy</thcudamemcpy<>	NVTX	update_halo [3,701 ms]	update_halo [3,672 ms]	update_halo [3,557 ms]	update_halo [3,689 ms]	update_halo [3,695 ms]	update_halo [3,688 ms]	upd
Profiler overhead 10 threads hidden + <td>CUDA API</td> <td>cudaMemcpy</td> <td>cudaMemcpy</td> <td>cudaMemcpy</td> <td>cudaMemcpy</td> <td>cudaMemcpy</td> <td>cudaMemcpy</td> <td>cud</td>	CUDA API	cudaMemcpy	cudaMemcpy	cudaMemcpy	cudaMemcpy	cudaMemcpy	cudaMemcpy	cud
10 threads hidden ← ↓ CUDA HW (0000:65:00.0 - NV > 99.7% Kernels devi	Profiler overhead							2.0
 CUDA HW (0000:65:00.0 - NV 99.7% Kernels 0.3% Memory 0.3% Memory MVTX update_nalo [1 update_nalo [3.673 ms] update_nalo [3.665 ms] update_nalo [3.691 ms] 	10 threads hidden 🗕 🕂							
> 99.7% Kernels devi devic devi devi devi devic devi devic devi devic devi	▼ CUDA HW (0000:65:00.0 - NV					Ť	i	11
▶ 0.3% Memory updat	▶ 99.7% Kernels	devi devic	devi devic	devi devic	devi devic	devi devic	devi devic	
NVTX updat update_halo [1,] update_halo [1,] update_halo [3,707 ms] update_halo [1,] update.halo [3,.01 ms] update.halo [3,.691 ms] update.hal	▶ 0.3% Memory							
 [52198] ./tea_leaf Threads (11) [52198] MPI Rank 0 - MPI MPI_Waitall [] MPI_Waitall []	NVTX	updat	pdate_halo [1,]	update_halo [1,	update_ha	lo [3,707 ms]	update_halo [1,	updat
Threads (11)	🕶 🔿 [52198] ./tea_leaf							
 V [52198] MPI Rank 0 MPI MPI_Waitall [MP	 Threads (11) 							
MPI MPI_Waitall [MPI_Waitall [MPI_Waitall [NVTX update_halo [3,704 ms] update_halo [3,673 ms] update_halo [3,682 ms] update_halo [3,565 ms] update_halo [3,691 ms] update_halo [3,691 ms]	✓ √ [52198] MPI Rank 0 -							
MP1 MP1_waitali [] MP1_waitali [] MP1_waitali [] NVTX update_halo [3,704 ms] update_halo [3,673 ms] update_halo [3,682 ms] update_halo [3,565 ms] update_halo [3,691 ms] update_halo [3,691 ms]								
NATY nhoare_uaio (2) to a unzi inhoare_uaio (2)	MP1	undate halo [2 704 ms]	undata bala [2.672 mc]	undata bala [2 692 mc]	undata hala 12 565 mcl	undata halo [2 601 mc]	undata balo I2 601 mcl	lunda
	INVIA	4	abaare_liaio [5,675 filis]	L abrace_liaio [2,002 [1]5]	T aboare_uaio [2/202 [U2]		7 abrare_uaio [2,631 iu2]	[upua

Sample from https://github.com/UK-MAC/TeaLeaf

Build TeaLeaf

Get the source code

git clone https://github.com/UK-MAC/TeaLeaf CUDA.git; cd TeaLeaf CUDA

Get Fortran NVTX module and apply provided TeaLeaf NVTX patch wget

https://raw.githubusercontent.com/maxcuda/NVTX_example/master/nvtx.f90
git apply /p/project/training2123/work/dietrich3/tea_nvtx.diff

Load the environment modules

module add Stages/2022 NVHPC OpenMPI CUDA Nsight-Systems
#source /p/project/training2123/work/schmitt5/load_modules.sh

Build TeaLeaf

if you did not apply the patch modify the Makefile: # add AMPERE architecture and -std=c++14 flag for nvcc make -e COMPILER=PGI

Alternative: use the pre-compiled binary with minimal NVTX instrumentation /p/project/training2123/work/dietrich3/TeaLeaf CUDA/tea leaf

Profile TeaLeaf

Run TeaLeaf without profiling

srun -A training2123 -p booster -N 1 --ntasks-per-node=2 --gres=gpu:2 -t 10
./tea_leaf

Profile with NVTX

srun -A training2123 -p booster -N 1 --ntasks-per-node=2 --gres=gpu:2 -t 10
\ nsys profile -t cuda,mpi,ucx,osrt,nvtx --mpi-impl=openmpi \
-y 1 -d 8 --kill=none -o tea-nvtx.%q{SLURM PROCID} --stats=true ./tea leaf

Profile with CPU and GPU sampling

```
srun -A training2123 -p booster -N 1 --ntasks-per-node=2 --gres=gpu:2 -t 10
\ nsys profile -t cuda,mpi,ucx,osrt,nvtx --mpi-impl=openmpi \
-y 1 -d 8 --kill=none -o tea-sampling%q{SLURM_PROCID} \
--backtrace=dwarf --sampling-period=3000000 \
--gpu-metrics-set=ga100 --gpu-metrics-device=0,1 \
--gpu-metrics-frequency=15000 \
./tea_leaf
```

Analyze the Profile

Investigate CLI stats output (--stats)

Open the report file in the Nsight Systems GUI

- 1. Start JupyterLab
- 2. Open Xpra Desktop
- 3. module load Nsight-Systems
- 4. nsys-ui REPORTFILE.nsys-rep
- 5. Add report of second MPI rank into timeline (via GUI: File -> Add report (beta)

