

## Nsight Compute CERN Compute Accelerator Forum

Felix Schmitt Mahen Doshi, Jonathan Vincent

### Nsight Product Family

#### **Workflow**

Nsight Systems - Analyze application algorithms system-wide <u>https://www.olcf.ornl.gov/calendar/nvidia-profiling-tools-nsight-systems/</u>

Nsight Compute - Analyze CUDA kernels

Nsight Graphics - Debug/analyze graphics workloads





CUDA Kernel profiler

Targeted metric sections for various performance aspects

Customizable data collection and presentation (tables, charts, ...)

UI and Command Line

Python-based rules for guided analysis (or post-processing)

Support for remote profiling across machines and platforms



#### Detailed memory workload analysis chart and tables



Comparison of results directly within the tool with "Baselines"

Supported across kernels, reports, and GPU architectures

Sour	rce: pm	e_spread.cu 🔽 🗖 Find			Source	e: pr	ne_spline_an	nd_spread_kernel 🔻 🕒 Find	~ ^	
Nav	igation:	Sampling Data (All)	18 18 €		Navig	ation	Sampling (	Data (All)	소 및 법 중	
	- #		Live Peristers Sampli	ing Data (All)		#	Address	Source	Live Peoisters	Samoling Data (All)
=	210	*/	Live Registers Sampli	ng Data (All)			Address	528 R4. SR TTD.7	Live Registers	
	211	<pre>if (atomIndexOffset &gt;= kernelParams.atoms.nAtoms)</pre>	10	26		10	0000200	S2R R3, SR TID.Y		11
	212			-1		11	0000200	S2B R2, SR TID-X		14
	213	return:		0		12	0000200	IMAD R7, R4, c[0x0][0x4], R3	14	s - 1
	214			0		13	0000200	IMAD R10, R7, c[0x0][0x0], R2	15	19
	215	/* Charges, required for both spline and spread */		0	= >	14	0000200	ISETP.GT.AND P0, PT, R10, 0xf, PT	15	2
	216	if (c_useAtomDataPrefetch)				15	0000200	@!P0 IADD3 R6, R0, R10, RZ	15	6
	217						0000200	@!P0 MOV R7, 0x4	14	
	218						0000200	<pre>@!P0 IMAD.WIDE R6, R6, R7, c[0x0][0x200]</pre>	14	13
>	219	pme_gpu_stage_atom_data <float, atomsperblock,<="" td=""><td>15</td><td>577</td><td></td><th>18</th><td>0000200</td><td>@!P0 LDG.E.SYS R12, [R6]</td><td>14</td><td>537</td></float,>	15	577		18	0000200	@!P0 LDG.E.SYS R12, [R6]	14	537
	220			Total Sample Coun	at: 577	9	0000200	ISETP.GT.AND P1, PT, R10, 0x2f, PT	12	0
	221	syncthreads();	11	0.17% Math Pipe T	Throttle (1	1) 0	0000200	@!P0 SHF.L.U32 R15, R10, 0x2, RZ	12	
	222	<pre>atomCharge = sm_coefficients[atomIndexLocal];</pre>	13	0.17% No Instructi	tions (1)	1	0000200	@!P1 MOV R9, 0x4	12	6
	223			0.17% Selected (1) 0.35% Dispatch St	) tall (2)	2	0000200	@!P1 IMAD R8, R5, 0x30, R10	12	
	224			1.73% Wait (10)		3	0000200	<pre>@!P1 IMAD.WIDE R8, R8, R9, c[0x0][0x1f8]</pre>	12	7
	225			1.91% Long Score	board (11	) 4	0000200	SHF.R.S32.HI R11, RZ, 0x1f, R10	13	
	226	atomCharge = kernelParams.atoms.d_coefficient:		2.08% Not Selecter	ed (12)	5	0000200		13	18
	227			8.32% Mio Throttle	le (48)	6	0000200	NOP	11	
	228			83.02% Lg Throttle	e (479)	7	0000200	BAR.SYNC 0x0	11	79
	229	<pre>if (computeSplines)</pre>				28	0000200	@!P1 LDG.E.SYS R13, [R8]	11	167
	230					29	0000200	LEA.HI R11, R11, R10, RZ, 0X5		
	231	<pre>if (c_useAtomDataPrefetch)</pre>				30	0000200	BSSY B0, 0x2000999532c0	و ا	
	232			e 🗌		31	0000200	LOP3.LUT R6, R4, 0x1, RZ, 0xc0, !PT	10	
	233	// Coordinates		e [		32	0000200	SHF.R.S32.HI R7, RZ, 0x5, R11	11	5
	234	sharedfloat_sm_coordinates[DIM * atom					0000200	@!P1 SHF.L.U32 R14, R10, 0x2, RZ	10	
	235			0		34	0000200	IMAD R10, R3, c[0x0][0x0], R2	10	

Source/PTX/SASS analysis and correlation

Source metrics per instruction and aggregated (e.g. PC sampling data)

Metric heatmap

## An example:

## GROMACS 2020 pme spread/gather kernels Old Version

Memory units more utilized than SM (Compute), but overall utilization is low Nsight Compute hints that this is a latency issue, recommends further sections to check We will still go through other sections for training purposes



Memory chart shows that stores are much more common in this kernel, transferring ~10x as much data as reads

Since bandwidth is not saturated, it's likely frequent operations



We have many active warps available, but most of them are not eligible (and hence not issued) on average

The next section (Warp State Statistics) can indicate which stall reasons cause this



Most important stall reason (by far) is LG (local/global) Throttle This indicates extremely frequent memory instructions, according to the guided analysis rule



Disabling global memory writes to store temporary data (for the gather kernel) could reduce this latency issue

This implies that the gather kernel has to re-compute this data

pn	ne_cal	culate_spli	nes.cuh 👻 ⊟		P+ stall_lg			- 1		, <b>i</b> , +
	#	Source				Sampling Data (All)	stall_lg			
	197		<pre>sm_fractCoords[share</pre>	edMemoryIndex]	= t - tInt;	18	0			
	198		<pre>tableIndex += tInt;</pre>			4	0			
	199		<pre>assert(tInt &gt;= 0);</pre>			0				
	200		<pre>assert(tInt &lt; c_pme</pre>	NeighborUnitcel	<pre>lCount * n);</pre>	/0				
	201					0				
	202		// TODO have shared	table for both	parameters t	c 0				
	203		<pre>// TODO compare text</pre>	ure/LDG perform	nance	G				
	204		sm_fractCoords[share	edMemoryIndex]	+=	0				
	205		fetchFromPar	amLookupTable(	kernelParams.	ç O				
	206				kernelParams.	f O				
	207		<pre>sm_gridlineIndices[s</pre>	sharedMemoryInd	ex] =	0				
	208		fetchFromPar	amLookupTable(	kernelParams.	( 0				
	209				ternelParans.	<del>;</del> 0				
	210		<pre>if (writeGlobal)</pre>			0				
	211		{			0				
	212		gm_gridlineIndic	es[atomIndex0f	fset * DIM +	s 282	224			
	213		sm_gridl	ineIndices[sha	redMemoryInde		Total Samp	le Count	282	
	214		}			1	Dispatch St	tall: 1 ( 0.	4%)	
	215	3					Lg Throttle	: 224 (79	.4%)	10()
	216						Mio Throttle	e: 43 (15	.2%)	4.70)
	217	/	* B-spline calculation	*/			Not Selecte	ed: 12 ( 4	.3%)	
	218						Selected: 1	(0.4%)		
_	219	9	const int chargeCheck =	pme_gpu_check_	atom_charge(	a 0				
	220	1	(chargeCheck)			G				

### gather: Old Version

#### More balanced compute/memory utilization, but also likely latency bound



### gather: Old Version

Reads temporary spline\_and\_spread kernel data from global memory Therefore, much more load operations and data transfered in that direction



#### gather: Old Version

#### Long Scoreboard stalls cause most wasted cycles These indicate waiting on local or global memory



## GROMACS 2020 pme spread/gather New Version

#### **Code Changes**

https://redmine.gromacs.org/projects/gromacs/repository/revisions/22118220401cee6f51d49c0a034e9fe5b4ba4260/diff?utf8=%E2 %9C%93&type=sbs

Two new template arguments added to spread/gather kernels Optimal kernel selected based on input data size Disabled temp data storage in global memory for this analysis

pme_spline_and_spread_kernel	pme_gather_kernel
<b>writeSplinesToGlobal</b>	<b>readGlobal</b>
control if we should write spline data to	control if we should read spline values
global memory	from global memory
useOrderThreadsPerAtom*	<i>useOrderThreadsPerAtom*</i>
control if we should use order or	control if we should use order threads per
order*order threads per atom	atom (order*order used if false)

\* not activated

### spline\_and\_spread: New Version

#### Overall performance improvement is ~15% (fewer cycles) Highest contributor appears to be the 54% reduced GPU DRAM throughput (SOL FB)



#### gather: New Version

#### Performance decreased slightly compared with "unoptimized" version The other individual sections would allow us to identify what has changed in detail



#### **New Version Summary**

#### Overall, combined performance improved by ~10% Use CSV export from CLI or UI to further analyze data in e.g. Excel

Function Name	Dema Proce: Device I Cycle	s [cycle] S	OL Memory [%]	SOL SM [%]	Function Name	Demanc Proce: Device Cycle	s [cycle]	SOL Memory [%]	SOL SM [%]
pme spline and spread kernel	voi… [17… Tesla…	83.249	49,04	20,33	pme spline and spread kernel	void … [13… Tesl…	69.516	44,82	21,65
pme gather kernel	voi… [17… Tesla…	29.646	53,38	50,84	pme gather kernel	void … [13… Tesl…	33.206	60,90	67,82
pme spline and spread kernel	voi… [17… Tesla…	83.236	49,06	20,32	pme spline and spread kernel	void … [13… Tesl…	68.969	45,18	21,84
pme gather kernel	voi… [17… Tesla…	29.890	53,01	50,37	pme gather kernel	void … [13… Tesl…	33.497	60,29	67,15
pme spline and spread kernel	voi… [17… Tesla…	83.624	48,80	20,23	pme spline and spread kernel	void … [13… Tesl…	68.516	45,44	21,97
pme gather kernel	voi… [17… Tesla…	29.051	54,58	51,86	pme gather kernel	void … [13… Tesl…	33.459	60,44	67,31
pme spline and spread kernel	voi… [17… Tesla…	83.617	48,80	20,23	pme spline and spread kernel	void … [13… Tesl…	69.197	45,00	21,75
pme gather kernel	voi… [17… Tesla…	29.636	53,52	50,81	pme gather kernel	void … [13… Tesl…	34.020	59,38	66,15
pme spline and spread kernel	voi… [17… Tesla…	83.070	49,12	20,37	pme spline and spread kernel	void … [13… Tesl…	70.337	44,29	21,39
pme gather kernel	voi… [17… Tesla…	29.591	53,68	50,87	pme gather kernel	void … [13… Tesl…	33.389	60,50	67,42
pme spline and spread kernel	voi… [17… Tesla…	83.235	49,05	20,32	pme spline and spread kernel	void … [13… Tesl…	69.615	44,73	21,62
pme gather kernel	voi… [17… Tesla…	29.705	53,20	50,67	pme gather kernel	void … [13… Tesl…	33.558	60,23	67,06
pme spline and spread kernel	voi… [17… Tesla…	82.910	49,19	20,40	pme spline and spread kernel	void … [13… Tesl…	69.261	44,96	21,73
pme gather kernel	voi… [17… Tesla…	29.233	54,32	51,50	pme gather kernel	void … [13… Tesl…	33.457	60,46	67,32
pme spline and spread kernel	voi… [17… Tesla…	83.109	49,09	20,35	pme spline and spread kernel	void … [13… Tesl…	68.927	45,16	21,84
pme gather kernel	voi… [17… Tesla…	29.634	53,61	50,79	pme gather kernel	void … [13… Tesl…	33.874	59,62	66,40
pme spline and spread kernel	voi… [17… Tesla…	83.299	48,98	20,31	pme spline and spread kernel	void … [13… Tesl…	69.266	44,95	21,73
pme gather kernel	voi… [17… Tesla…	30.130	52,62	49,97	pme gather kernel	void … [13… Tesl…	33.221	60,91	67,82
pme spline and spread kernel	voi… [17… Tesla…	83.595	48,84	20,24	pme spline and spread kernel	void … [13… Tesl…	68.452	45,50	22,00
pme gather kernel	voi… [17… Tesla…	29.554	53,74	50,99	pme gather kernel	void … [13… Tesl…	33.432	60,54	67,32

## Back to theory: Nsight Compute on your cluster

### **Collecting Data**

By default, CLI results are printed to stdout
Use --export/-o to save results to a report file, use -f to force overwrite
\$ ncu -f -o \$HOME/my\_report <app>
\$ my\_report.ncu-rep

Use --log-file to pipe text output to a different stream (stdout/stderr/file)

Can use (env) variables available in your batch script or file macros to add report name placeholders Full parity with nvprof filename placeholders/file macros \$ ncu -f -o \$HOME/my\_report\_%h\_\${LSB\_JOBID}\_%p <app> \$ my\_report\_host01\_951697\_123.ncu-rep

https://docs.nvidia.com/nsight-compute/NsightComputeCli/index.html#command-line-options-filemacros



On a single-node submission, one Nsight Compute instance can profile all launched child processes

Data for all processes is stored in one report file

ncu --target-processes all -o <singlereport-name> <app> <args>



On multi-node submissions, one tool instance can be used per node

### Ensure that instances don't write to the same report file on a shared disk

ncu -o report\_%q{OMPI\_COMM\_WORLD\_RANK}
<app> <args>



Multiple tool instances on the same node are supported, but...

All kernels across all GPUs will be serialized using system-wide file lock

All instances must be of the same user, or have file permissions (fixed in next version - permissions are set automatically)

fi



Consider profiling only a single rank, e.g. using a wrapper script

```
#!/bin/bash
if [[ "$OMPI_COMM_WORLD_RANK" == "3" ]] ; then
    /sw/cluster/cuda/11.1/ nsight-compute/ncu -
o report_${OMPI_COMM_WORLD_RANK} --target-
processes all $*
else
    $*
```

#### What To Collect

#### Curated "sets" and "sections" with commonly-used, high-value metrics

\$ ncu --list-sets
Identifier Sections

Estimated Metrics

default	LaunchStats, Occupancy, SpeedOfLight	35
detailed	ComputeWorkloadAnalysis, InstructionStats, LaunchStats, MemoryWorkloadAnaly	157
	sis, Occupancy, SchedulerStats, SourceCounters, SpeedOfLight, SpeedOfLight_	
	RooflineChart, WarpStateStats	
full	ComputeWorkloadAnalysis, InstructionStats, LaunchStats, MemoryWorkloadAnaly	162
	sis, MemoryWorkloadAnalysis_Chart, MemoryWorkloadAnalysis_Tables, Occupancy	
	, SchedulerStats, SourceCounters, SpeedOfLight, SpeedOfLight_RooflineChart,	
	WarpStateStats	
		4 7

source SourceCounters

47

#### Use defaults, or combine as desired

\$ ncu --set default --section SourceCounters --metrics sm\_\_inst\_executed\_pipe\_tensor.sum ./my-app

#### What To Collect

#### Query metrics for any targeted chip

\$ ncu --query-metrics --chip ga100 smsp warps issue stalled not selected cumulative # of warps waiting for the microscheduler to select the warp to issue smsp warps issue stalled selected cumulative # of warps selected by the microscheduler to issue an instruction smsp warps issue stalled short scoreboard cumulative # of warps waiting for a scoreboard dependency on MIO operation other than (local, global, surface, tex) tpc cycles active # of cycles where TPC was active # of cycles where TPC was active tpc cycles elapsed ==PROF== Note that these metrics must be appended with a valid suffix before profiling them. See --help for more information on --query-metrics-mode.

#### Specify sub-metrics in section files, or on the command line

```
$ ncu --query-metrics-mode suffix --metrics sm__inst_executed_pipe_tensor ./my-app
sm__inst_executed_pipe_tensor.sum
sm__inst_executed_pipe_tensor.avg
sm__inst_executed_pipe_tensor.min
```

### Source Analysis

SASS (assembly) is always available embedded into the report CUDA-C (Source) and PTX availability depends on compilation flags Use -lineinfo to include source/SASS correlation data in the binary

```
cmake/gmxManageNvccConfig.cmake:201
macro(GMX_SET_CUDA_NVCC_FLAGS)
    set(CUDA_NVCC_FLAGS "${GMX_CUDA_NVCC_FLAGS};${CUDA_NVCC_FLAGS};-lineinfo")
endmacro()
```

Source is not embedded in the report, need local access to the source file to resolve in the UI Comparing different iterations (e.g. optimizations) of the same source file can be difficult

Next version will support importing CUDA-C source into the report

Compiler optimizations can prevent exact source/SASS correlation

## Another example:

Roofline analysis for image feature extraction kernel



ExtractFeatures analyzes an image for interesting pixels

Part of the algorithm is converting the pixel color format

Kernel is heavily compute-bound, utilizing the SM units > 80%

Fp64 (64bit/double floating-point math) pipeline is by far the biggest contributor



Roofline chart suggests to use float instead of double, due to the 32:1 peak perf ratio between 32bit and 64bit HW units (on Turing)

On V100, the ratio would be 2:1

vigation: Instructions Executed 🔹 👻 🛆 🐼	te te 주	Navigation:	Instructions Executed	• • ^	소 믭 막 소
# Source	Instructions Executed		Source	Sampling Data (All)	Sampling Data (Not Issued) 🔺
<pre>10_ glDisable(GL_DEPTH_TEST);</pre>		120	ISETP.LT.OR P3, PT, R10,	18	14
10_		121	ISETP.GE.OR P3, PT, R5, o	12	9 _ [
10_ // viewport		122	P3 LDG.E.SYS R10, [R16+0xc]	70	62
<pre>10_ glViewport(0, 0, window_width, window_height);</pre>			LOP3.LUT R24, R18, Øxff,		0
10_		124	LOP3.LUT R20, R20, 0xff,	9	5
10_ SDK_CHECK_ERROR_GL();		125	LOP3.LUT R15, R15, Øxff,		
10_		126	I2F.F64.U16 R18, R24	12,771	12,249
10_ return true;		127	LOP3.LUT R27, R27, Øxff,		9
10_ }		128	I2F.F64.U16 R20, R20	8,698	8,352
10_		129	LOP3.LUT R25, R25, Øxff,		
10_ #endif // !NO_GL		130	I2F.F64.U16 R22, R15	11,650	11,186
10_		131	IADD3 R5, R5, 0x1, RZ	2	2 -
10_ // ********************* Worker (server) thread ********		132	I2F.F64.U16 R16, R27	11,520	11,029
10_		133	I2F.F64.U16 R24, R25	2,200	2,122
10_ // convert an RGB(A) color to its luminance value (gr		134	I2F.F64.U32 R14, R14	10,001	9,623
10device unsigned char RGBtoLuminance(		135	DMUL R18, R18, c[0x2][0x0	2,630	2,526
10_ const uchar4& color	=	136	DFMA R18, R20, c[0x2][0x8	10,517	10,098
10_ )	Ξ	137	DMUL R20, R16, c[0x2][0x0	9,300	8,961
10 {		138	DFMA R18, R22, c[0x2][0x:	9,277	8,895
10_ return (FLOAT_T)0.299 * color.x + (FLOAT_T)0.587	307,330,880	139	LOP3.LUT R22, R26, 0xff,	510	484
10_ }		140	I2F.F64.U16 R22, R22	9,160	8,824
10_		141	F2I.U32.F64.TRUNC R18, R1	2,436	2,848
10device uchar4 GetPixel(		142	P0 SHF.R.U32.HI R19, RZ, RZ,	428	401
10_ uchar4* pImg,		143	DFMA R20, R22, c[0x2][0x8	9,428	9,060
10_ int x,		144	P0 SHF.R.U32.HI R22, RZ, 0x8	458	416
10_ int y,		145	LOP3.LUT R26, R18, Øxff,		
10_ int imgw,		146	PRMT R18, RZ, 0x7610, R18	9	3 💌

On the Source page, we can see where in the code those instructions are executed.

Make sure that "Instructions Executed" is selected in the metrics drop down, then use navigation buttons.

# Live

Current		241 - ExtractFeatures (	992, 992,	Time: 4.49 msecond	Cycles: 5,978,351	Regs: 42	GPU: GeForce RTX 2080 Ti	SM Frequency:	1.33 cycle/nsecond	CC: 7.5	Process: [2334	] imgserver_	3 🕀 \ominus
Baseline	1	241 - ExtractFeatures (	992, 992,	Time: 60.17 msecond	Cycles: 81,344,906	Regs: 47	GPU: GeForce RTX 2080 Ti	SM Frequency:	1.35 cycle/nsecond	CC: 7.5	Process: [2113	] imgserver	2
▼ GPU Spe	eed Of Light										F	1	- Q
- High lovel our	ioniou of the i	utilization for compute ar	d momoriu roco	urses of the CDU. For en	ch unit the Speed Of L	ight (COL) ro	parts the achieved percenter	o of utilization with	respect to the theor	otical may	imum. High lawalu	uon iou of i	the utilization
for compute a	and memory re	sources of the GPU pres	ented as a roof	line chart.	ch unit, the speed Of L	light (SOL) re	ports the achieved percentag	e or utilization with	respect to the theor	eucaimax	mum. nign-level (	verview or t	me utilization
SOL SM [%]					93.75 (	(+8.46%) D	ouration [msecond]					4.49	(-92.53%)
SOL Memory	/ [%]				14.98 (+1,2	269.27%) E	lapsed Cycles [cycle]					5,978,351	(-92.65%)
SOL L1/TEX	Cache [%]				29.97 (+1,2	269.27%) S	M Active Cycles [cycle]				5,9	55,653.59	(-92.51%)
SOL L2 Cach	:he [%]				3.30 (+1,1	106.42%) S	M Frequency [cycle/nsec	ond]				1.33	(-1.64%)
SOL DRAM [5	[%]				7.64 (+1,1	101.37%) D	RAM Frequency [cycle/ns	econd]				1.67	(-0.74%)
						GPU Utili	ization						
										1			
SM [9	%]												
Memory [9	%1												
	· P											<u> </u>	
	0.0	10.0	20.0	30.0	40.0		50.0 60.0		70.0	80.0	90	0	100.0
						Speed	d Of Light [%]						
			SOL SM Bre	akdown				50	Memory Break	rdown			
								00	- Memory Brear	uom			
SOL SM	/: Inst Execut	ed Pipe Xu [%]			93.75 (+692,3	360.47%)	SOL L1: Data Pipe Lsu Wa	vefronts [%]				14.98 (+1	,269.27%)
SOL SM	/I: Pipe Alu Cy	cles Active [%]		61.54 (+1,262.78%)								14.01 ( . 1	250.0500
SOL SM	I: Issue Active	SOL SM: Pipe Alu Cycles Active [%]			01134(11)5	262.78%)	SOL L1: Lsuin Requests [9	6]				14.01 (+1	,259.05%)
SOL SM	SOL SM: Issue Active [%]				52.85 (+1,2	262.78%) 260.55%)	SOL L1: Lsuin Requests [9 SOL L1: Lsu Writeback Ac	6] tive [%]				9.89 (+1	,259.05%) ,266.48%)
SOL SM: Inst Executed [ /6]					52.85 (+ 1,2 52.85 (+ 1,2	262.78%) 260.55%) 258.98%)	SOL L1: Lsuin Requests [9 SOL L1: Lsu Writeback Ac SOL GPU: Dram Through	6] tive [%] put [%]				9.89 (+1 7.64 (+1	,259.05%) ,266.48%) ,101.37%)
SOL SM	∕I: Inst Executi ∕I: Pipe Fma C	e [%] ed [%] Sycles Active [%]			52.85 (+ 1,2 52.85 (+ 1,2 15.99 (+ 8,2	262.78%) 260.55%) 258.98%) 249.91%)	SOL L1: Lsuin Requests [? SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads	6] tive [%] put [%] [%]				9.89 (+1) 7.64 (+1) 3.31 (+1)	,259.05%) ,266.48%) ,101.37%) ,261.38%)
SOL SM SOL SM	/l: Inst Executo /l: Pipe Fma C /l: Inst Executo	e [%] ed [%] Sycles Active [%] ed Pipe Lsu [%]			52.85 (+ 1,2 52.85 (+ 1,2 52.85 (+ 1,2 15.99 (+ 8,2 14.02 (+ 1,2	260.55%) 258.98%) 249.91%) 260.65%)	SOL L1: Lsuin Requests [? SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: T Sectors [%]	6] tive [%] put [%] [%]				9.89 (+ 1 7.64 (+ 1 3.31 (+ 1 3.30 (+ 1	,259.05%) ,266.48%) ,101.37%) ,261.38%) ,106.42%)
SOL SM SOL SM SOL SM	Л: Inst Executo Л: Pipe Fma C Л: Inst Executo Л: Mio Inst Iss	e [%] ed [%] :ycles Active [%] ed Pipe Lsu [%] ued [%]			52.85 (+ 1,2 52.85 (+ 1,2 52.85 (+ 1,2 15.99 (+ 8,2 14.02 (+ 1,2 4.68 (+	262.78%) 260.55%) 258.98%) 249.91%) 260.65%) • 18.72%)	SOL L1: Lsuin Requests [? SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: T Sectors [%] SOL L2: Xbar2lts Cycles A	6] tive [%] put [%] [%] ctive [%]				9.89 (+1 7.64 (+1 3.31 (+1 3.30 (+1 2.14 (+1	,259.05%) ,266.48%) ,101.37%) ,261.38%) ,106.42%) ,141.25%)
SOL SM SOL SM SOL SM SOL SM	Л: Inst Executo Л: Pipe Fma C Л: Inst Executo Л: Mio Inst Iss Л: Mio2rf Writ	e [%] ed [%] Sycles Active [%] ed Pipe Lsu [%] ued [%] teback Active [%]			52.85 (+ 1,2 52.85 (+ 1,2 15.99 (+ 8,2 14.02 (+ 1,2 4.68 (+ 3.59 (+ 1,2	262.78%) 260.55%) 258.98%) 249.91%) 260.65%) • 18.72%) 265.71%)	SOL L1: Lsuin Requests [? SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: T Sectors [%] SOL L2: Xbar2lts Cycles A SOL L2: Lts2xbar Cycles A	6] tive [%] put [%] [%] ctive [%] ctive [%]				9.89 (+1 7.64 (+1 3.31 (+1 3.30 (+1 2.14 (+1 1.79 (+1	,259.05%) ,266.48%) ,101.37%) ,261.38%) ,106.42%) ,141.25%) ,000.57%)
SOL SM SOL SM SOL SM SOL SM SOL SM	Л: Inst Executo Л: Pipe Fma C Л: Inst Executo Л: Mio Inst Iss Л: Mio2rf Writ Л: Inst Executo	e [%] ed [%] ed Pipe Lsu [%] ued [%] teback Active [%] ed Pipe Cbu Pred On A	vny [%]		52.85 (+ 1,2 52.85 (+ 1,2 15.99 (+ 8,2 14.02 (+ 1,2 4.68 (+ 3.59 (+ 1,2 0.60 (+ 1,2	202.78%) 260.55%) 258.98%) 249.91%) 260.65%) 18.72%) 265.71%) 260.26%)	SOL L1: Lsuin Requests [9 SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: T Sectors [%] SOL L2: Xbar2lts Cycles A SOL L2: Lts2xbar Cycles A SOL L2: D Sectors [%]	6] tive [%] put [%] [%] ctive [%] ctive [%]				9.89 (+1 7.64 (+1, 3.31 (+1, 3.30 (+1, 2.14 (+1, 1.79 (+1, 1.45 (+1,	,259.05%) ,266.48%) ,101.37%) ,261.38%) ,106.42%) ,141.25%) ,000.57%) ,289.63%)
SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM	A: Inst Executo A: Pipe Fma C A: Inst Executo A: Mio Inst Iss A: Mio2rf Writ A: Inst Executo A: Mio Pq Rea	e [%] ed [%] ed Pipe Lsu [%] ued [%] ieback Active [%] ed Pipe Cbu Pred On A id Cycles Active [%]	\ny [%]		52.85 (+1,2 52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.60 (+1,2 0.31 (+1,2	262,78%) 260,55%) 258,98%) 249,91%) 260,65%) 18,72%) 265,71%) 265,26%) 256,96%)	SOL L1: Lsuin Requests [9 SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: TSectors [%] SOL L2: Xbar2lts Cycles A SOL L2: LtS2xbar Cycles A SOL L2: D Sectors [%] SOL L1: M L1tex2xbar Red	6] tive [%] out [%] [%] ctive [%] q Cycles Active [%]	1			9.89 (+1 7.64 (+1) 3.31 (+1) 3.30 (+1) 2.14 (+1) 1.79 (+1) 1.45 (+1) 1.33 (+1)	,259.05%) ,266.48%) ,101.37%) ,261.38%) ,106.42%) ,141.25%) ,000.57%) ,289.63%) ,274.78%)
SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM	A: Inst Executo A: Pipe Fma C A: Inst Executo A: Mio Inst Iss A: Mio2rf Writ A: Inst Executo A: Mio Pq Rea A: Mio Pq Writ	e [%] ed [%] iycles Active [%] ed Pipe Lsu [%] ued [%] ieback Active [%] ed Pipe Cbu Pred On A id Cycles Active [%] te Cycles Active [%]	vny [%]		52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.60 (+1,2 0.31 (+1,2 0.31 (+1,2	262.78%) 260.55%) 258.98%) 249.91%) 260.65%) -18.72%) 265.71%) 260.26%) 256.96%) -97.25%)	SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: T Su Writeback Ac SOL L2: T Sectors [%] SOL L2: T Sectors [%] SOL L2: Xbar2lts Cycles A SOL L2: L152xbar Cycles A SOL L2: D Sectors [%] SOL L1: M L1tex2xbar Ret SOL L1: M Xbar2l1tex Rea	6] tive [%] Jut [%] [%] ctive [%] ctive [%] q Cycles Active [%] d Sectors [%]				9.89 (+1 7.64 (+1 3.31 (+1, 3.30 (+1, 2.14 (+1, 1.79 (+1, 1.45 (+1, 1.33 (+1, 1.16 (+1,	,259.05%) ,266.48%) ,101.37%) ,261.38%) ,106.42%) ,141.25%) ,000.57%) ,289.63%) ,274.78%) ,277.26%)
SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM	A: Inst Executo A: Pipe Fma C A: Inst Executo A: Mio Inst Iss A: Mio2rf Writ A: Inst Executo A: Mio Pq Rea A: Mio Pq Writ A: Inst Executo	e [%] ed [%] ed Pipe Lsu [%] ued 7%] seback Active [%] ed Pipe Cbu Pred On A Id Cycles Active [%] te Cycles Active [%] ed Pipe Adu [%]	ıny [%]		52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.60 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+	262.78%) 260.55%) 258.98%) 249.91%) 260.65%) -18.72%) 265.71%) 260.26%) 256.96%) -97.25%) 261.38%)	SOL L1: Lsuin Requests [3 SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: TSectors [%] SOL L2: Xba2ltS Cycles A SOL L2: L5Xbar Cycles A SOL L2: L5Xbar Cycles A SOL L1: M L1texZbar Ret SOL L1: M Xba2l1tex Rea SOL L2: D Sectors Fill Des	6] tive [%] [%] ctive [%] ctive [%] q Cycles Active [% d Sectors [%] ice [%]				9.89 (+1 7.64 (+1 3.31 (+1, 3.30 (+1, 2.14 (+1, 1.79 (+1, 1.45 (+1, 1.33 (+1, 1.16 (+1, 1.11 (+1)	,259,05%) ,266,48%) ,101,37%) ,261,38%) ,106,42%) ,141,25%) ,000,57%) ,289,63%) ,274,78%) ,277,26%) ,007,52%)
SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM	A: Inst Execut A: Pipe Fma C A: Inst Execut A: Mio Inst Iss A: Mio2rf Writ A: Inst Execut A: Mio Pq Rea A: Mio Pq Wri A: Inst Execut A: Inst Execut	e [%] ked [%] ycles Active [%] ded Pipe Lsu [%] ued [%] seback Active [%] de Pipe Cbu Pred On A d Cycles Active [%] te Cycles Active [%] ed Pipe Adu [%] ed Pipe Adu [%]	sny [%]		52.85 (+1,2 52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.60 (+1,2 0.31 (+1,2 0.31 (- 0.03 (+1,2 0.03 (+1,2) 0.02 (+1,2	262.78%) 260.55%) 258.98%) 249.91%) 260.65%) -18.72%) 265.71%) 260.26%) 260.26%) 256.96%) -97.25%) 261.38%) 261.38%)	SOL L1: Lsu M Requests [ SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: TSectors [%] SOL L2: Xbar2lts Cycles A SOL L2: D Sectors [%] SOL L1: M LTeex2tbar Read SOL L1: M Xbar2litex Read SOL L2: D Sectors Fill Dev SOL L2: D Sectors Fill Dev SOL L2: D Sectors Fill Dev SOL L2: D Sectors Fill Dev	6] tive [%] [%] [%] ctive [%] ctive [%] q Cycles Active [%] d Sectors [%] ice [%] i]				9.89 (+1 7.64 (+1 3.31 (+1) 3.30 (+1) 2.14 (+1) 1.79 (+1) 1.45 (+1) 1.16 (+1) 1.11 (+1) 1.11 (+1)	,259,05%) ,266,48%) ,101.37%) ,261.38%) ,106,42%) ,141.25%) ,000.57%) ,289,63%) ,274,78%) ,277,26%) ,007.52%) ,007.52%)
SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM	A: Inst Execute A: Pipe Fma C A: Inst Execute A: Mio Inst Iss A: Mio2rf Writ A: Inst Execute A: Mio Pq Rea A: Mio Pq Wri A: Inst Execute A: Inst Execute C: Request Cy	e [%] ed [%] ycles Active [%] ed Pipe Lsu [%] ed Pipe Lsu [%] ed Pipe Cbu Pred On A d Cycles Active [%] d Cycles Active [%] ed Pipe Adu [%] ed Pipe Uniform [%] ycles Active [%]	ıny [%]		52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.60 (+1,2 0.31 (+ 0.03 (+1,2 0.03 (+1,2 0.02 (+1,2 0.02 (+1,2 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.03 (+1	262.78%) 260.55%) 258.98%) 258.98%) 249.91%) 260.65%) 18.72%) 260.26%) 260.26%) 256.96%) -97.25%) 261.38%) 261.38%) 261.38%) 261.38%)	SOL L1: Lsu Mrequests [3 SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: T Sectors [%] SOL L2: Xbar2lts Cycles A SOL L2: L152xbar Cycles A SOL L2: D Sectors [%] SOL L1: M Xbar2/1tex Rea SOL L1: M Xbar2/1tex Rea SOL L2: D Sectors Fill Dev SOL L2: D Sectors Fill Dev SOL L2: D at Bank Writes	6) tive [%] (%) ctive [%] ctive [%] q Cycles Active [%] d Sectors [%] ice [%] i3] [%]				9.89 (+1 7.64 (+1 3.31 (+1 3.30 (+1, 2.14 (+1, 1.79 (+1, 1.45 (+1, 1.16 (+1, 1.11 (+1, 1.10 (+1, 1.05 (+1, 0.27 (+1)	,259,05%) ,266,48%) ,101.37%) ,261.38%) ,106,42%) ,141.25%) ,000.57%) ,289,63%) ,274,78%) ,277,26%) ,007.52%) ,044.35%) ,272,66%)
SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SD SOL SD	A: Inst Execute A: Pipe Fma C A: Inst Execute A: Mio Inst Iss A: Mio2rf Writ A: Inst Execute A: Mio Pq Rea A: Mio Pq Rea A: Mio Pq Writ A: Inst Execute C: Request Cy A: Inst Execute C: Inst Execute	e [%] ed [%] ycles Active [%] ed Pipe Lsu [%] ued [%] teback Active [%] ed Pipe Cbu Pred On A di Cycles Active [%] ed Pipe Adu [%] ed Pipe Uniform [%] ycles Active [%] ed Pipe Fol f %]	uny [%]		52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+1,2) 0.31 (+1,2) (+1,2) 0.31 (+1,2)	262.78%) 260.55%) 258.98%) 258.98%) 260.65%) 18.72%) 260.26%) 260.26%) 256.96%) 97.25%) 261.38%) 261.38%) 261.38%) + 0.00%) + 0.00%)	SOL L1: Lsuin Requests [9 SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: T Sectors [%] SOL L2: Xbazlts Cycles A SOL L2: Lt52xbar Cycles A SOL L2: D Sectors [%] SOL L1: M L1tex2xbar Red SOL L1: M Xbazl1tex. Red SOL L1: M Skazl1tex. Red SOL L2: D Sectors Fill Dev SOL L2: T Tag Requests [9 SOL L1: Data Bank Writes SOL L2: D Chamic Input C	6) tive [%] [%] ctive [%] ctive [%] q Cycles Active [%] ice [%] 5] [%] (%] voles Active [%]				9.89 (+1 7.64 (+1 3.31 (+1 3.30 (+1, 2.14 (+1, 1.79 (+1, 1.45 (+1, 1.33 (+1, 1.16 (+1, 1.11 (+1, 1.05 (+1, 0.27 (+1, 0.00 (+1)	,259,05%) ,266,48%) ,101,37%) ,261,38%) ,106,42%) ,141,25%) ,000,57%) ,274,78%) ,277,26%) ,007,52%) ,004,35%) ,272,68%) ,272,68%) ,260,57%)
SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SM SOL SOL SOL SM SOL SM	A: Inst Execute A: Pipe Fma C A: Inst Execute A: Mio Inst Iss A: Mio Inst Iss A: Mio Pq Writ A: Inst Execute A: Inst Execute A: Inst Execute C: Request C) A: Inst Execute A: Inst Execute	e [%] ycles Active [%] ed Pipe Lsu [%] ued [%] seback Active [%] ed Pipe Cbu Pred On A id Cycles Active [%] ed Pipe Adu [%] ed Pipe Hofform [%] ycles Active [%] ed Pipe [76] [%]	vny [%]		52.85 (+1,2 52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.60 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+1,2 0.03 (+1,2 0.03 (+1,2 0.02 (+1,2 0 ( 0 ( 0 ( 0 ( 0 (	262.78%) 260.55%) 258.98%) 249.91%) 260.65%) 118.72%) 265.71%) 260.26%) 260.26%) 265.96%) 97.25%) 261.38%) 261.38%) 261.38%) 261.38%) 261.38%) 260.26%	SOL L1: Lsu M Requests [ SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: TSectors [%] SOL L2: Xbar2lts Cycles A SOL L2: D Sectors [%] SOL L1: M LTeex2tar Read SOL L2: D Sectors [%] SOL L1: M Xbar2litex Read SOL L2: D Sectors Fill Dev SOL L2: D Avenic Input C SOL L2: D Atomic Input C	6) tive [%] (%) ctive [%] ctive [%] ctive [%] ctive [%] d Sectors [%] d Sectors [%] (%) [%] (%] ycles Active [%]				9.89 (+1 7.64 (+1 3.31 (+1 3.30 (+1 2.14 (+1, 1.79 (+1, 1.45 (+1, 1.33 (+1, 1.16 (+1, 1.11 (+1) 1.05 (+1, 0.00 (+1, 0.00 (+1)	229.05%) 266.48%) 101.37%) 261.38%) 106.42%) 1141.25%) 000.57%) 289.63%) 274.78%) 277.26%) 007.52%) 004.35%) 272.68%) 2261.38%)
SOL SM SOL SM	4: Inst Execute 4: Pipe Fma C 4: Inst Execute 4: Mio Inst Iss 4: Mio2rf Writ 4: Inst Execute 4: Mio Pq Rea 4: Mio Pq Rea 4: Inst Execute 4: Inst Inst Inst Inst 4: Inst Inst 4: Inst Inst 4: Inst Inst 4: Inst 4: Inst Inst 4: Inst	e [%] ed [%] yytes Active [%] ed Pipe Lsu [%] ed Pipe Lsu [%] ed Pipe Cbu Pred On A d Cycles Active [%] ed Pipe Cbu Pred On A d Cycles Active [%] ed Pipe Uniform [%] ycles Active [%] ed Pipe Fp16 [%] ed Pipe Ipa [%] ed Pipe [pa [%]	ny [%]		52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+1,2 0.03 (+1,2 0.03 (+1,2 0.02 (+1,2 0.02 (+1,2) 0.02 (-1,2) 0 ( 0 ( 0 ( 0 ( 0 ( 0 (	162.78%) 160.55%) 160.655%) 1249.91%) 1260.65%) 18.72%) 165.71%) 166.26%) 197.25%) 1261.38%) 1261.38%) 1261.38%) 1261.38%) 1261.38%) 1261.38%) 1261.38%) 1260.00	SOL L1: Lsu Mrequests [3 SOL L1: Lsu Mreback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: T Sectors [%] SOL L2: DSectors [%] SOL L2: DSectors [%] SOL L1: M L1: M L1: AC SOL L2: D Sectors [%] SOL L1: M L1: M L2: T Sol Requests [9 SOL L2: D ata Bank Writes SOL L2: D Atomic Input C SOL L1: F Wavefronts [%] SOL L1: F Wavefronts [%]	6) tive [%] vut [%] ctive [%] ctive [%] d Sectors [%] ice [%] ig [%] ycles Active [%] .Cycles Active [%]				9.89 (+1 3.31 (+1 3.31 (+1 3.30 (+1) 2.14 (+1 1.79 (+1 1.45 (+1) 1.33 (+1) 1.33 (+1) 1.33 (+1) 1.35 (+1) 0.07 (+1 0.00 (+1) 0.00 (+1)	,259.05%) ,266.48%) ,101.37%) ,261.38%) ,104.42%) ,111.25%) ,000.57%) ,289.63%) ,277.26%) ,007.52%) ,007.52%) ,007.52%) ,044.35%) ,272.68%) ,270.68%) ,261.38%)
SOL SM SOL SM	4: Inst Execute 4: Pipe Fma C 4: Inst Execute 4: Mio Inst Iss 4: Mio21 Writ 4: Inst Execute 4: Mio Pq Rea 4: Mio Pq Rea 4: Inst Execute 4: Inst Execute 4: Inst Execute 4: Inst Execute 4: Inst Execute 4: Inst Execute 4: Pipe FoE4 C	e [%] ed [%] yoles Active [%] ed Pipe Lsu [%] ed Pipe Lsu [%] ed Pipe Cbu Pred On A di Cycles Active [%] ed Pipe Active [%] ed Pipe Active [%] ed Pipe Uniform [%] rcles Active [%] ed Pipe [%] ed Pipe [%] ed Pipe [%]	\ny [%]		52.85 (+1,2 52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+1,2 0.31 (+1,2 0.03 (+1,2 0.03 (+1,2 0.02 (+1,2 0 0 ( 0 ( 0 ( 0 ( 0 ( 0 ( 0 ( 0	<pre>bc.78%) bc.75%) bc.55%) bc.95%) bc.95%) bc.75%) bc.75%) bc.75%) bc.75%) bc.77%) bc.26%) bc.26%) bc.26%) bc.25%) bc.25%) bc.02%) bc.00%) bc.00%) bc.00%) bc.00%) bc.00%) bc.00%)</pre>	SOL L1: Lsu Mrequests [3 SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: T Sectors [%] SOL L2: Xbar2lts Cycles A SOL L2: L52Xbar Cycles A SOL L2: D Sectors [%] SOL L1: M L1tex2Xbar Ret SOL L1: N bar2l1ter Rea SOL L2: D Sectors Fill Dev SOL L2: D Sectors Fill Dev SOL L2: Tag Requests [9 SOL L2: D Atomic Input C SOL L1: D Atomic Input C SOL L1: F Wavefronts [%] SOL L1: Fill Spie Jier Wa	6) tive (%) vut (%) (%) (%) ctive (%) q Cycles Active (%) d Sectors (%) ice (%) (%) ycles Active (%) (Cycles Active (%) (Cycles Active (%)	1			9.89 (+1 3.31 (+1 3.30 (+1 2.14 (+1) 1.45 (+1) 1.33 (+1) 1.45 (+1) 1.45 (+1) 1.16 (+1 1.11 (+1) 1.05 (+1) 0.027 (+1) 0.00 (+1) 0.0	259.05%) 266.48%) 101.37%) 261.38%) 106.42%) 1141.25%) 000.57%) 289.63%) 274.78%) 277.26%) 007.52%) 004.35%) 272.68%) 260.37%) 261.38%) (±0.00%)
SOL SM SOL SM	4: Inst Execute 4: Inst Execute 4: Inst Execute 4: Mio Inst Iss 4: Mio2rf Writ 4: Mio Pq Rea 4: Mio Pq Rea 4: Mio Pq Rea 4: Inst Execute 4: Pipe Fp64 0 4: Pipe Sharea	e [%] ed [%] ycles Active [%] ed Pipe Lsu [%] ued [%] teback Active [%] ed Pipe Cou Pred On A id Cycles Active [%] te Cycles Active [%] ed Pipe Active [%] ed Pipe Inform [%] ycles Active [%] ed Pipe Tex [%] ed Pipe Tex [%] Cycles Active [%]	vny [%]		52.85 (+1,2 52.85 (+1,2 52.85 (+1,2 52.85 (+1,2 15.99 (+8,2 14.02 (+1,2 4.68 (+ 3.59 (+1,2 0.60 (+1,2 0.31 (+1,2 0.03 (+1,2 0.03 (+1,2 0.02 (+1,2 0.02 (+1,2 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.03 (+1,2) 0.02 (+1,2) 0.03 (+1,2) 0.02 (+1,2) 0.03 (+1,2) 0.02 (+1,2) 0.02 (+1,2) 0.03 (+1,2) 0.02 (+1,2) 0.0	102.78%) 102.78%) 1258.98%) 1269.55%) 1269.55%) 1260.55%) 1260.55%) 1260.26%) 12	SOL L1: Lsu M Requests [ SOL L1: Lsu Writeback Ac SOL GPU: Dram Through SOL L1: Data Bank Reads SOL L2: T Sectors [%] SOL L2: Xbar2lts Cycles A SOL L2: D Sectors [%] SOL L1: M LTecx2bar Read SOL L2: D Sectors [%] SOL L1: M Xbar2litex. Read SOL L2: D Sectors [%] SOL L1: D Tecx2bar Read SOL L2: D Sectors [%] SOL L1: Data Bank Writes SOL L2: D Atomic Input C SOL L1: Texin Sm2tex. Read SOL L1: Data Pipe Tex. Wa SOL L1: Data Pipe Tex. Wa SOL L1: Data Pipe Tex. Wa	6) tive [%] val [%] (%) ctive [%] ctive [%] d Sectors [%] ice [%] [%] ycles Active [%] i Cycles Active [%] i Cycles Active [%]	1			9.89 (+1 3.31 (+1 3.31 (+1 3.30 (+1 2.14 (+1) 1.79 (+1) 1.45 (+1) 1.45 (+1) 1.45 (+1) 1.45 (+1) 1.45 (+1) 1.45 (+1) 0.00 (+1) 0.00 (+1) 0 0	229U5%) 26648%) 26648%) 26138%) 10137%) 26138%) 10142%) 114125%) 00057%) 28963%) 27726%) 000752%) 004335%) 27726%) 00752%) 004335%) 27726%) 26138%) 27126%) 26138%) 26138%) 26138%) 26138%) 26138%) 26138%) 26138%) 26138%) 27126%) 26138%)

Change the kernel to use 32bit-precision math, only.

Can now compare all data from the first, unoptimized kernel compared against the Current (blue) kernel.

Most notably, it is 92% faster (Duration), and doesn't use 64bit (Fp64) math anymore.

Since we are blocked less by the fp64 pipeline, we now have a 12x better memory HW-units utilization

### **Further Reading**

https://docs.nvidia.com/nsight-compute/ProfilingGuide/index.html#roofline

https://github.com/NVIDIA/nsight-training/tree/master/cuda/2020\_gtc

https://gitlab.com/NERSC/roofline-on-nvidia-gpus

## Conclusion

#### **Known Issues/Outlook**

https://docs.nvidia.com/nsight-compute/ReleaseNotes/index.html#known-issues

#### Outlook for next version

Source import

Roofline analysis improvements

Source-level divergence metrics

Profile series experiments

#### Conclusion

Nsight Compute enables detailed kernel analysis

Rules give guidance on optimization opportunities and help metric understanding

Limit sections/metrics to what is required when overhead is a concern

Still requires level of hardware understanding to fully utilize the tool - pay attention to rule results and refer to <a href="https://docs.nvidia.com/nsight-compute/ProfilingGuide/index.html">https://docs.nvidia.com/nsight-compute/ProfilingGuide/index.html</a>



## THANK YOU!

Download	https://developer.nvidia.com/nsight-compute (can be newer than toolkit version)
Documentation	https://docs.nvidia.com/nsight-compute (and local with the tool)
Forums	https://devtalk.nvidia.com
Further Training	https://developer.nvidia.com/nsight-compute-videos https://developer.nvidia.com/nsight-compute-blogs



#### PME spread/gather



https://indico.cern.ch/event/791712/contributions/3407944/attachments/1839813/3015919/Tk1.3-MolecularDynamicsGromacs-F2FCERN.pdf