

GPU ACCELERATORS AT JSC SUPERCOMPUTING INTRODUCTION COURSE

13 November 2024 | Andreas Herten | Forschungszentrum Jülich



Member of the Helmholtz Association

Outline

GPUs at JSC JUWELS JUWELS Cluster JUWELS Booster JURECA DC JUPITER **GPU** Architecture **Empirical Motivation** Comparisons **GPU** Architecture Summary

Programming GPUs Libraries Directives CUDA C/C++ Performance Analysis Advanced Topics Advanced Topics





JUWELS Cluster – Jülich's Scalable System

- = 2500 nodes with Intel Xeon CPUs (2 \times 24 cores)
- 46 + 10 nodes with 4 NVIDIA Tesla V100 cards (16 GB memory)
- 10.4 (CPU) + 1.6 (GPU) PFLOP/s peak performance (Top500: #86)





JUWELS Booster – Scaling Higher!

- 936 nodes with AMD EPYC Rome CPUs (2 \times 24 cores)
- Each with 4 NVIDIA A100 Ampere GPUs (each: FP64TC: 19.5 FP64TC: 19.5 TFLOP/S, 40 GB memory)
- InfiniBand DragonFly+ HDR-200 network; 4×200 Gbit/s per node







Top500 List Nov 2020:

- #1 Europe
- #7 World
- #4* Top/Green500



JUWELS Booster – Scaling Higher!

- = 936 nodes with AMD EPYC Rome CPUs (2 \times 24 cores)
- Each with 4 NVIDIA A100 Ampere GPUs (each: FP64TC: 19.5 FP64: 9.7 TFLOP/S, 40 GB memory)
- InfiniBand DragonFly+ HDR-200 network; 4 \times 200 Gbit/s per node





JURECA DC – Multi-Purpose

- 768 nodes with AMD EPYC Rome CPUs (2 \times 64 cores)
- 192 nodes with 4 NVIDIA A100 Ampere GPUs
- InfiniBand DragonFly+ HDR-100 network





JUPITER – Exascale

- First Exascale system in Europe
- Procured by EuroHPC JU, BMBF, MKW-NRW, hosted by JSC
- Currently in pre-installation
- 24 000 NVIDIA H100 GPUs (Grace-Hopper superchips)
- 1 EFLOP/S FP64 (HPL), 32 EFLOP/S FP8 (peak)
- \rightarrow jupiter.fz-juelich.de

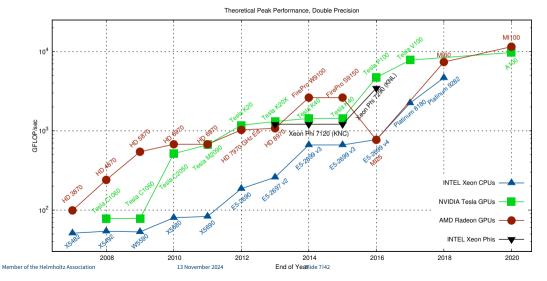




GPU Architecture

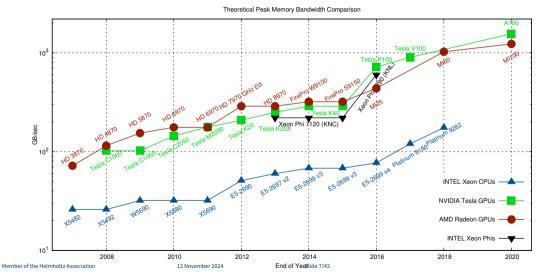
Status Quo Across Architectures

Performance



Status Quo Across Architectures

Memory Bandwidth



CPU vs. GPU

A matter of specialties







CPU vs. GPU

A matter of specialties



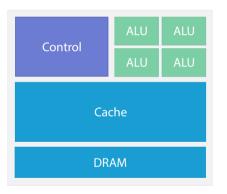
Transporting one



Transporting many



CPU vs. GPU _{Chip}





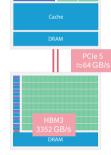


GPU Architecture Design

GPU optimized to hide latency

- Memory
 - GPU has small (40 GB), but high-speed memory 1555 GB/s
 - Stage data to GPU memory: via PCIe 4 (32 GB/s) or PCIe 5 (64 GB/s) bus
 - Stage automatically (Unified Memory), or manually
- Two engines: Overlap compute and copy





Host

Device









H100 80 GB RAM, 3352 GB/s



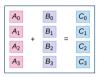
13 November 2024

Slide 10142



CPU:

- Single Instruction, Multiple Data (SIMD)
- Simultaneous Multithreading (SMT)
- GPU: Single Instruction, Multiple Threads (SIMT)
 - CPU core \cong GPU multiprocessor (SM)
 - Working unit: set of threads (32, a warp)
 - Fast switching of threads (large register file)
 - Branching if _____

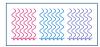


Vector

SMT



SIMT





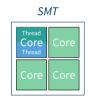
SIMT

$\mathsf{SIMT}=\mathsf{SIMD}\oplus\mathsf{SMT}$



Vector





Graphics: img:amperepictures

SIMT





Slide 11 42

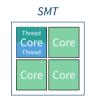
SIMT

$\mathsf{SIMT}=\mathsf{SIMD}\oplus\mathsf{SMT}$



Vector





Graphics: img:amperepictures

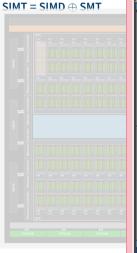
SIMT





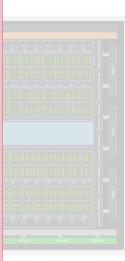
Slide 11 42

SIMT



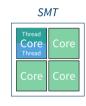
Multiprocessor





Vector





SIMT





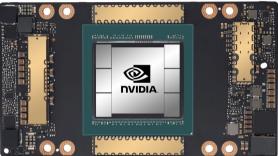
13 November 2024

Slide 11 42

A100 vs H100

Comparison of last vs. current generation

A100



H100





A100 vs H100

Comparison of last vs. current generation





A100 vs H100

Comparison of last vs. current generation

A100

SM	71200		
L1 Instruction Cache			
L0 Instruction Castle			
Warg Scheduler (32 thread/c8)	Werp Scheduler (32 Ihreadicik)		
Dispatch Unit (32 thread/clk)	Dispatch Unit (32 threadictk)		
Register File (16,384 x 32-58)	Register File (16,264 x 32-bit)		
ACT31 ACT31 FP33 FP33 FP34	NTL NTL 1722 1722 1754		
ACT33 ACT32 FP32 FP32 FP34	NTRA NTRA 1722 1722 1754		
ACT31 ACT31 FP33 FP33 FP34	NTL NTL 1722 1722 1754		
NT31 NT31 PP31 P332 PM4	NTRA NTRA 1722 1722 1754		
ATTS ATTS FPN FPN FPN FPN	NT2 NT2 FF22 FF22 FF54 TENSOR CORE		
ACT31 ACT31 FF53 FF53 FF54	NT12 NT12 FF22 FF22 FF94		
ACT31 ACT31 FF53 FF53 FF54	NTE NTE 172 172 173		
8/33 8/31 PP32 PP32 PP44	NT11 NT11 1733 1732 1794		
1위 1위 1위 1위 1위 1위 1위 1위 1위 1	· · · · · · · · · · · · · · · · · · ·		
LD institution Carba			
LD Instruction Cache LD Instruction Cache Ward Scientizer (22 Unreally IS) Ward Scientizer (22 Unreally IS)			
Dispatch Unit (22 thread/c8) Dispatch Unit (22 thread/c8)			
	Register File (16,384 x 32-bit)		
8/33 8/33 PESS PESS F/9/4	672 672 773 773 774		
8/33 8/33 FP32 FP32 FP34	NT21 NT21 1722 1722 1744		
8/33 8/33 PP52 PP52 PP64	NT2 NT2 172 172 173 174		
NT32 NT32 PP32 PP32 PP44 TENSOR CORE	1172 1172 1722 1722 1794 TENSOR CORE		
8/33 6/33 FP32 FP32 FP34	1012 10121 1722 1732 1732 1744		
NITSS NITSS FERSE FERSE	N121 N121 FP22 FP32 FP44		
8/33 8/33 FP32 FP32 FP34	N122 N122 FP22 FP22 FP34		
NUTRA BUTTA FIRMA FIRMA	NT22 NT22 FF22 FF22 FF94		
다. 나가 나가 나가 나가 나가 나가 나가 동두U			
	ische / Shared Herrory		

H100





CPU vs. GPU

Let's summarize this!



Optimized for low latency

- + Large main memory
- + Fast clock rate
- + Large caches
- + Branch prediction
- + Powerful ALU
- Relatively low memory bandwidth
- Cache misses costly
- Low performance per watt



Optimized for high throughput

- + High bandwidth main memory
- + Latency tolerant (parallelism)
- + More compute resources
- + High performance per watt
- Limited memory capacity
- Low per-thread performance
- Extension card



Programming GPUs

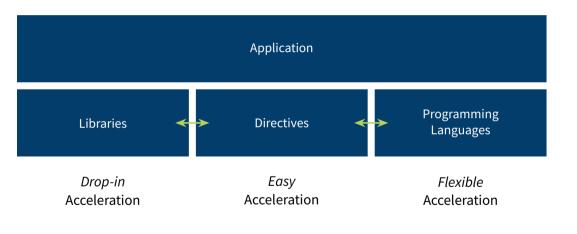
Preface: CPU

A simple CPU program!

```
SAXPY: \vec{y} = a\vec{x} + \vec{y}, with single precision
Part of LAPACK BLAS Level 1
void saxpy(int n, float a, float * x, float * y) {
  for (int i = 0; i < n; i + +)
    v[i] = a * x[i] + v[i]:
}
int a = 42:
int n = 10:
float x[n], v[n];
// fill x, y
saxpy(n, a, x, y);
```

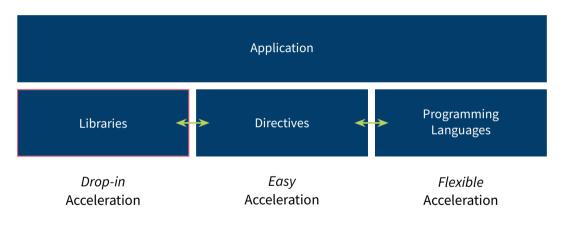


Summary of Acceleration Possibilities





Summary of Acceleration Possibilities







Programming GPUs is easy: Just don't!





Programming GPUs is easy: Just don't!

Use applications & libraries





Programming GPUs is easy: Just don't!

Use applications & libraries





Libraries

Programming GPUs is easy: Just don't!

Use applications & libraries



Л ПСН

Forschungszentrum

Libraries

Programming GPUs is easy: Just don't!

Use applications & libraries



Л ПСН

Forschungszentrum



- GPU-parallel BLAS (all 152 routines)
- Single, double, complex data types
- Constant competition with Intel's MKL
- Multi-GPU support
- → https://developer.nvidia.com/cublas http://docs.nvidia.com/cuda/cublas



cuBLAS

Code example

```
int a = 42; int n = 10;
float x[n], y[n];
// fill x. v
cublasHandle t handle;
cublasCreate(&handle):
float * d x, * d y;
cudaMallocManaged(\delta d x, n * sizeof(x[0])):
cudaMallocManaged(\delta d y, n * sizeof(y[0]));
cublasSaxpv(handle. n. a. d x. 1. d v. 1):
cublasGetVector(n, sizeof(y[0]), d_y, 1, y, 1);
cudaFree(d x); cudaFree(d y);
cublasDestroy(handle);
```



cuBLAS

Code example

int a = 42; int n = 10;

<pre>float x[n], y[n]; // fill x, y</pre>	
cublasHandle_t handle; cublasCreate(&handle);	Initialize
float * d_x, * d_y; cudaMallocManaged(&d_x, n * sizeof(x[0]));● cudaMallocManaged(&d_y, n * sizeof(y[0]));	Allocate GPU memory
cublasSaxpy(handle, n, a, d_x, 1, d_y, 1);●	Call BLAS routine
<pre>cublasGetVector(n, sizeof(y[0]), d_y, 1, y, 1);</pre>	Copy result to host
cudaFree(d_x);	Finalize



Programming GPUs Directives

GPU Programming with Directives

Keepin' you portable

Annotate serial source code by directives

#pragma acc loop
for (int i = 0; i < 1; i++) {};</pre>



GPU Programming with Directives

Keepin' you portable

Annotate serial source code by directives

```
#pragma acc loop
for (int i = 0; i < 1; i++) {};</pre>
```

- OpenACC: Especially for GPUs; OpenMP: Has GPU support
- Compiler interprets directives, creates according instructions



GPU Programming with Directives

Keepin' you portable

Annotate serial source code by directives

```
#pragma acc loop
for (int i = 0; i < 1; i++) {};</pre>
```

- OpenACC: Especially for GPUs; OpenMP: Has GPU support
- Compiler interprets directives, creates according instructions

Pro

- Portability
 - Other compiler? No problem! To it, it's a serial program
 - Different target architectures from same code
- Easy to program

Con

- Only few compilers
- Not all the raw power available
- A little harder to debug



OpenACC / OpenMP

Code example

```
void saxpy_acc(int n, float a, float * x, float * y) {
    #pragma acc kernels
    for (int i = 0; i < n; i++)
        y[i] = a * x[i] + y[i];
}
float a = 42;
int n = 10;
float x[n], y[n];
// fill x, y
saxpy_acc(n, a, x, y);</pre>
```



OpenACC / OpenMP

Code example

```
void saxpy_acc(int n, float a, float * x, float * y) {
    #pragma omp target map(to:x[0:n]) map(tofrom:y[0:n]) loop
    for (int i = 0; i < n; i++)
        y[i] = a * x[i] + y[i];
}
float a = 42;
int n = 10;
float x[n], y[n];
// fill x, y
saxpy_acc(n, a, x, y);</pre>
```



Programming GPUs CUDA C/C++

Finally...



Finally...

OpenCL Open Computing Language by Khronos Group (Apple, IBM, NVIDIA, ...) 2009

- Platform: Programming language (OpenCL C/C++), API, and compiler
- Targets CPUs, GPUs, FPGAs, and other many-core machines
- Fully open source



Finally...

OpenCL Open Computing Language by Khronos Group (Apple, IBM, NVIDIA, ...) 2009

- Platform: Programming language (OpenCL C/C++), API, and compiler
- Targets CPUs, GPUs, FPGAs, and other many-core machines
- Fully open source

CUDA NVIDIA's GPU platform 2007

- Platform: Drivers, programming language (CUDA C/C++), API, compiler, tools, ...
- Only NVIDIA GPUs
- Compilation with nvcc (free, but not open) clang has CUDA support, but CUDA needed for last step
- Also: CUDA Fortran; and more in NVIDIA HPC SDK



Finally...

OpenCL Open Computing Language by Khronos Group (Apple, IBM, NVIDIA, ...) 2009

- Platform: Programming language (OpenCL C/C++), API, and compiler
- Targets CPUs, GPUs, FPGAs, and other many-core machines
- Fully open source

CUDA NVIDIA's GPU platform 2007

- Platform: Drivers, programming language (CUDA C/C++), API, compiler, tools, ...
- Only NVIDIA GPUs
- Compilation with nvcc (free, but not open)
 - clang has CUDA support, but CUDA needed for last step
- Also: CUDA Fortran; and more in NVIDIA HPC SDK
- HIP AMD's unified programming model for AMD (via ROCm) and NVIDIA GPUs 2016+
- SYCL Intel's unified programming model for CPUs and GPUs (also: DPC++)



Finally...

OpenCL Open Computing Language by Khronos Group (Apple, IBM, NVIDIA, ...) 2009

- Platform: Programming language (OpenCL C/C++), API, and compiler
- Targets CPUs, GPUs, FPGAs, and other many-core machines
- Fully open source

CUDA NVIDIA's GPU platform 2007

- Platform: Drivers, programming language (CUDA C/C++), API, compiler, tools, ...
- Only NVIDIA GPUs
- Compilation with nvcc (free, but not open)
 - clang has CUDA support, but CUDA needed for last step
- Also: CUDA Fortran; and more in NVIDIA HPC SDK
- HIP AMD's unified programming model for AMD (via ROCm) and NVIDIA GPUs 2016+

SYCL Intel's unified programming model for CPUs and GPUs (also: DPC++)

- Choose what flavor you like, what colleagues/collaboration is using
- Hardest: Come up with parallelized algorithm



Finally...

OpenCL Open Computing Language by Khronos Group (Apple, IBM, NVIDIA, ...) 2009

- Platform: Programming language (OpenCL C/C++), API, and compiler
- Targets CPUs, GPUs, FPGAs, and other many-core machines
- Fully open source

CUDA NVIDIA's GPU platform 2007

- Platform: Drivers, programming language (CUDA C/C++), API, compiler, tools, ...
- Only NVIDIA GPUs
- Compilation with nvcc (free, but not open)
- clang has CUDA support, but CUDA needed for last step
- Also: CUDA Fortran; and more in NVIDIA HPC SDK
- HIP AMD's unified programming model for AMD (via ROCm) and NVIDIA GPUs 2016+

SYCL Intel's unified programming model for CPUs and GPUs (also: DPC++)

- Choose what flavor you like, what colleagues/collaboration is using
- Hardest: Come up with parallelized algorithm



In software: Threads, Blocks

Methods to exploit parallelism:



- Methods to exploit parallelism:
 - Thread



- Methods to exploit parallelism:
 - Threads





In software: Threads, Blocks

Methods to exploit parallelism:





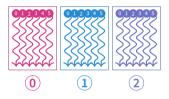


- Methods to exploit parallelism:
 - Threads \rightarrow Block
 - Block



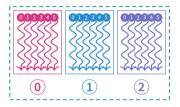


- Methods to exploit parallelism:
 - Threads \rightarrow Block
 - Blocks



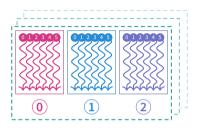


- Methods to exploit parallelism:
 - Threads \rightarrow Block
 - $\blacksquare \quad \mathsf{Blocks} \to \mathsf{Grid}$



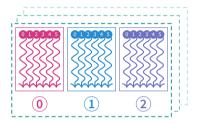


- Methods to exploit parallelism:
 - Threads \rightarrow Block
 - $\blacksquare \quad \mathsf{Blocks} \to \mathsf{Grid}$
 - Threads & blocks in 3D





- Methods to exploit parallelism:
 - Threads \rightarrow Block
 - $\blacksquare \quad \mathsf{Blocks} \to \mathsf{Grid}$
 - Threads & blocks in 3D
- Parallel function: kernel
 - __global__ kernel(int a, float * b) { }
 - Access own ID by global variables threadIdx.x, blockIdx.y,...
- Execution entity: threads
 - Lightweight → fast switchting!
 - 1000s threads execute simultaneously \rightarrow order non-deterministic!





CUDA SAXPY

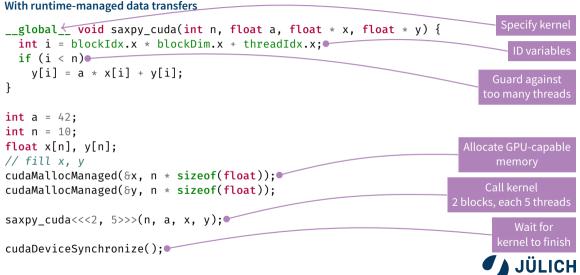
With runtime-managed data transfers

```
global void saxpy cuda(int n, float a, float * x, float * y) {
 int i = blockIdx.x * blockDim.x + threadIdx.x;
 if (i < n)
   v[i] = a * x[i] + v[i]:
}
int a = 42:
int n = 10:
float x[n]. v[n]:
// fill x. v
cudaMallocManaged(&x, n * sizeof(float));
cudaMallocManaged(&y, n * sizeof(float));
saxpv cuda<<<2. 5>>>(n. a. x. v):
```

```
cudaDeviceSynchronize();
```



CUDA SAXPY



Programming GPUs Performance Analysis

GPU Tools

The helpful helpers helping helpless (and others)

NVIDIA

cuda-gdb GDB-like command line utility for debugging compute-sanitizer Check memory accesses, race conditions, ... Nsight IDE for GPU developing, based on Eclipse (Linux, OS X) or Visual Studio (Windows) or VScode Nsight Systems GPU program profiler with timeline Nsight Compute GPU kernel profiler

AMD

rocProf Profiler for AMD's ROCm stack uProf Analyzer for AMD's CPUs and GPUs



Nsight Systems

•••

	ofilestats=tru Statistics:	e ./poisson	2d 10 # (shor	tened)		
Time(%) 90.9	Total Time (ns) 160,407,572	Num Calls 30	Average 5,346,919.1	Minimum 1,780	Maximum 25,648,117	Name cuStreamSynchronize
CUDA Kern	el Statistics:					
Time(%)	Total Time (ns)	Instances	Average	Minimum	Maximum	Name
100.0 0.0	158,686,617 25,120	10 10	15,868,661.7 2,512.0	14,525,819 2,304	25,652,783 3,680	main_106_gpu main_106_gpured



Nsight Systems

GUI

• • •			NVIDIA N	sight Systems 202	20.4.1								
<u>File</u> <u>View</u>	<u>File View Tools Help</u>												
report2.qd	report2.qdrep ×												
📑 Timelin	ne View 👻		D 1x O										
	• 18s 19s 20s 21s 22s 23s 24												
+ CPU (25	i6)												
Threads	(6)												
- CUDA H	W (Tesla V100-PCIE-160												
▼ 1.5% Ke	ernels												
▶ 79.6%	% main_106_gpu												
▶ 19.0%	% main_118_gpu												
3 ken	rnel groups hidder - +												
· 98.5%		ana debieteratis, debieteratis	densistiet.atest	nini, Airidense die	than i di jan i di	inte a professional a philes	naranti i Jaharanti shini yani shinin jihinin						
<0.1%	% Memset	aki jeliteteli, jelitetelij	ananisia ana	enina, disintenen ale	nerokiski, jantiski	ine sidilateni John	يتغافل والمعتقار أيتمر إفادا المعتقر بالمعتق						
<0.1% 51.1%	% Memset & HtoD memcpy						يتفاطر والاستغارةين والتواطعين والمعادي والمعادي والمعاد معادلة المالية المعادة المات سالة المتدامية المالية						
<0.1% 51.1%	% Memset	ولفا فساهرت المتقدرات الراقب	nation adda	and on the state of a	فالفعة بلدالفده	land, set of the same same states							
<0.1% 51.1%	% Memset 6 HtoD memcpy % DtoH memcpy	ولفا فساهرت المتقدرات الراقب	nation adda	and on the state of a	فالفعة بلدالفده	land, set of the same same states	وبالغر والمعادر أغتر والخر اغتضرت أغتاه						
<0.1% 51.1%	% Memset 6 HtoD memcpy % DtoH memcpy	i ah i da barina () da ba	nation adda	and on the state of a	فالفعة بلدالفده	land, set of the same same states	nt tai y tai a da d						
<0.1% 51.1% 48.9%	% Memset 6 HtoD memcpy % DtoH memcpy	i ah i da barina () da ba	nation adda	and on the state of a	فالفعة بلدالفده	land, set of the same same states	nt tai y tai a da d						
<0.1% 51.1% 48.9%	% Memset 6 HtoD memcpy % DtoH memcpy	and a share of a second s	nation adda	and on the state of a	فالفعة بلدالفده	valde L dist des lés les des a	ni tali (1904 tali yila) yila (aki yalanki yila) ni hali, imahalmati shafmatinali yiman I						
<0.1% 51.1% 48.99 Events Vie	% Memset 4 HtoD memcpy % DtoH memcpy	and a share of a second s	and tob [sh] as on tab	andra a transfer	nde taki siki sika da aki aki Inda aki na aki siki na	ind wishind of the indiana	atal (ustal)de salutati (ustal)de a salutati (ustal)de						
<0.1% 51.1% 48.9% Events View	% Memset & HtoD memcpy % DtoH memcpy aw	and a polyation of a set and	metickijskim ochiek medicilnostn, soched	palen i i si standa da da se	nd tot just, dank i da indination estatution GPU	net stal set set () she and she and she and she Name Context	atal (ustal)de salutati (ustal)de a salutati (ustal)de						
<0.1% 51.1% 48.9% Events Vie # 7	% Memset 4 HtoD memcpy % DtoH memcpy www Name Memset	und Jahlandal (Jahlandal) ad Jahlandal (Jahlandal) a	vatal)ste stal matanhan, adha ▼ Start 17,9516s	Duration 1,760 µs	GPU GPU 0	Name Context Stream 14	atal (ustal)de salutati (ustal)de a salutati (ustal)de						
<0.1% 51.1% 48.99 Events Viet # 7 8	% Memset	ut data a farancia (contrast a far bal data a farancia (contrast a far bal data a farancia (contrast a farancia (c	 Start 17,9516s 17,9516s 17,9517s 	Duration 1,760 µs 472,923 µs	GPU GPU 0 GPU 0	Name Context Stream 14 Stream 14	atal (ustal)de salutati (ustal)de a salutati (ustal)de						

Nsight Compute

GUI

-level overview of the utilization for compute and memory resources of the GPU. For each unit, the Speed Of Light (SoL) reports the achieved percentage of utilization with respect to the refect anximum. High-level overview of the utilization for compute and memory resources of the GPU presented as a nonline chart.	Launch: - m	nc_polymer_ite	eration_352	2_gpu * 🛛 🗑	Add Ba	iseline 💌 Ap	oly <u>R</u> ules							Save as Im	age	
Super Of Light N development of the utilization for compute and memory resources of the OPU, For each unit, the Speed Of Light (SOL) reports the achieved parcentage of utilization with respect to the tetral maximum. High-level conview of the utilization for compute and memory resources of the OPU, For each unit, the Speed Of Light (SOL) reports the achieved parcentage of utilization with respect to the GPU presented as a rooffine chart. Shi {\si} 20. 51 (+12.6%) 22.6.21 (+12.6%) 22.8.12 (+12.6%) 22.8.	94 Time: 25.82 n	nsecond Cyc	cles: 28.19	91.300 Reg	s: 144 GF	U: A100-SX	44-40GB	SM Frequer	ncy: 1.09 cy	cle/nsecond	CC: 8.0	Process: [31938] SO	MA 🕀 🛛	Э	
Interformation of the utilization for compute and memory resources of the GPU, for each unit, the Speed Of Light (SQ) reports the software direction with respect to the relation with respect to the utilization for compute and memory resources of the GPU prevented as a rootilic edur. SP (s)	6 Time: 77.17 m	nsecond Cyc	cles: 101.1	23.609 Reg	5: 144 GF	U: Tesla V10	0-SXM2-16GE	3 SM Frequer	ncy: 1.31 cy	cle/nsecond	CC: 7.0	Process: [30412] SOI	AN		
etical maximum. High-level overview of the utilization for compute and memory resources of the GPU presented as a rooflike char. Hemory [k]	l Of Light 🛕												AII	*	۶	
Memory (%) 55.82 (+1.26%) Elapsed Cycles [cycle] 2810308 (-7.2 L2 Cache (%) 61.89 (+4.26%) %friewerky (cycle) (cycle) 227784365.46 (-7.2 L2 Cache (%) 61.89 (+34.96%) %friewerky (cycle) (cycle) 227784365.46 (-7.2 L2 Cache (%) 61.89 (+34.96%) %friewerky (cycle) (cycle) 227784365.46 (-7.2 BM (%) 01.92 (-2.8.2%) DMM Frequency (cycle)rescond) 1.22 (-3.8 00 10,0 20,0 30,0 40,0 50,0 60,0 70,0 80,0 90,0 SOLS M Breakdown SOL Memory Breakdown <td colspa<="" td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>ercentage o</td><td>of utilization</td><td>with respe</td><td>ct to the</td><td></td></td>	<td></td> <td>ercentage o</td> <td>of utilization</td> <td>with respe</td> <td>ct to the</td> <td></td>											ercentage o	of utilization	with respe	ct to the	
LLT/EX 49.19 (+9.49) 94 Active (cycles [cycle] 27784365.46 -77 LLT/EX Cache [N] 61.89 (+9.490) 94 Active (cycles [cycle] 27784365.46 -77 LLT/EX Cache [N] 51.82 (+2.800) BMM Frequency [cycle/nsecond] 1.21 (-98 DRM [N] 51.92 (+2.800) BMM Frequency [cycle/nsecond] 1.22 (-98 M [N] 50.0 50.0 60.0 70.0 80.0 90.0 Solut Breakdown 50.0 50.0 60.0 70.0 80.0 90.0 Solut SM Instance Minite [N] 20.61 (+100.09%) 50.12 The Part Instance Minite [N] 558.2 (+12.40 Solut SM Instance Minite [N] 20.61 (+100.09%) 50.12 The Part Instance Minite [N] 558.2 (+12.40 Solut SM Instance Minite [N] 20.61 (+100.09%) 50.12 The Part Instance Minite [N] 558.2 (+12.40 Solut SM Instance/Inf [N] 20.61 (+100.09%) 50.12 The Part Instance [N] 558.2 (+12.40 Solut SM Instance/Inf [N] 20.61 (+100.09%) 50.12					20.51	L (+160.099) Duration	[msecond]					25.82	(-66.5	549	
L2 Cache (s) 61.89 (+349.81%) 9f / Frequency (cycle/nsecond) 1.99 (-1.9) DRAH (s) 31.32 (+2.82%) DRAH (s) 1.21 (-38 GPU Utilization GPU Utilization 1.21 (-38 M (s) 50.0 (6.0 70.0 80.0 90.0 90.0 50.0 (6.0 70.0 80.0 90.0 50.0 50.0 60.0 70.0 80.0 90.0 50.0 50.0 50.0 50.0 50.0 50.0 5	[26]				55.82	2 (+1.269) Elapsed	Cycles [cyc]	le]				28191300	(-72.1	12*	
DRAM [%] 31.52 (-42.82%) DRAM Prequency (cycLe/nsecond) 1.21 (-38 SM PG GPU Utilization	Cache [%]				40.19) (+5.499) SM Activ	e Cycles [c	ycle]			277	84365.64	(-72.3	309	
GPU Utilization GPU Utilization M (5) Sols M breakdown Sols M breakdown Sols M Breakdown Sols (+160.0%) Sols M Breakdown Sols M Breakdown Sols M Breakdown Sols M Breakdow	e [%]				61.89	0 (+149.019) SM Frequ	ency [cycle,	/nsecond]				1.09	(-16.6	57%	
SM (%) Solution <]				31.52	2 (-42.829) DRAM Fre	quency [cyc]	le/nsecon	3)			1.21	(+38.2	21%	
Immory [5] 0.0 10,0 20,0 30,0 40,0 50,0 60,0 70,0 80,0 90,0 Speed Of Light [5] SOL SM Breakdown SOL SM Breakdown SOL SM Breakdown SOL SM Ereakdown SOL SM Ereakdown [54] SOL SM Ereakdown [56] SOL SM Ereakdown [56] <t< td=""><td></td><td></td><td></td><td></td><td></td><td>GPU</td><td>Utilization</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>						GPU	Utilization									
norry [3] 0,0 10,0 20,0 30,0 40,0 50,0 60,0 70,0 80,0 90,0 Speed Of Light (%) SOL SM Breakdown 0.5 M: tost Source (%) 20,5 (+160,00%) 0.5 M: tost Source (%) 20,			_													
0.0 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0 SOL SM Breakdown SOL SM Breakdown OL SM Issue Active [%] SOL Marce State Sta																
0.0 10.0 20.0 20.0 20.0 40.0 50.0 60.0 70.0 80.0 90.0 SOL SM Breakdown SOL Marce Sol SM Breakdown SOL SM Breakdown SOL SM Breakdown SOL Marce Sol SM Breakdown SOL SM Breakdown <td col<="" td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td>	<td></td>															
0.0 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0 SOL SM Breakdown SOL SM Breakdown OL SM Issue Active [%] SOL Marce State Sta																
No. Speed of Uight (%) No. No. SOLS M Breakdown SOL SM Breakdown SOL Memory Breakdown 0.5 M: Instructure (%) 20.51 (160.00%) SOL L2: Margins (%) 65.82 (124.41) 0.6 M: Instructure (%) 20.46 (160.00%) SOL L2: Margins (%) 65.82 (124.41) 0.6 M: Instructure (%) 13.83 (166.60%) SOL L2: Tag Requests (%) 65.82 (124.41) 0.6 M: Instructure (%) 13.83 (166.60%) SOL L2: Tag Requests (%) 30.86 (171.11) 0.6 M: Instructure (%) 13.83 (166.60%) SOL L2: Tag Requests (%) 30.86 (171.11) 0.6 M: Instructure (%) 13.83 (166.60%) SOL L2: Tag Requests (%) 30.86 (171.11) 0.6 M: Instructure (%) 13.83 (171.82%) SOL UPU. Dam Trouppingt (%) 33.80 (171.11) 0.6 M: Instructure (%) 13.97 (191.93) SOL L2: D Sectors (%) 24.88 (493.10) 0.6 M: More Read Cycles Active (%) 8.10 (101.99%) SOL L2: D Sectors (%) 12.10 (12.10) 0.6 M: More Read Cycles Active (%) 72.21 (165.69%) SOL L1: Luiw Request (%) 10.89 (172.10) 0.1 SM: More Read Cycles Active (%) 65.85 (172.26%) SOL L2: L1.22.24 Sec Cycle																
No. Operator Speed of Uight (%) Operator Operator SOLS SM Breakdown SOLL SW breakdown SOLL SW breakdown SOLMemory Breakdown SOLS SM Interactive [%] 20,51 (1680,09%) SOLL 2: Marr21th Cycles Active [%] 6552 (124,41 SOLS SM Interactive [%] 20,46 (1480,10%) SOLL 2: Marr21th Cycles Active [%] 6552 (124,41 SOLS SM: Inter Executed [%] 20,46 (1480,10%) SOLL 2: Tag Requests [%] 6552 (124,41 SOLS SM: Inter Executed [%] 13,86 (-1680,69%) SOLL 2: Tag Requests [%] 3806 (-171,10%) SOLS SM: Inter Executed [%] 13,86 (-1680,69%) SOLL 2: Tag Requests [%] 3806 (-171,10%) SOLS SM: Inter Executed [%] 13,80 (-1680,69%) SOLL 2: Tag Requests [%] 3806 (-171,10%) SOLS SM: Inter Executed [%] 10,80 (-172,96%) SOLL 2: Tag Requests [%] 3806 (-171,10%) SOLS SM: Inter Executed [%] 10,80 (-172,96%) SOLL 2: Das Fore IND (%) 3830 (-171,10%) SOLS SM: Inter Executed [%] 8,10 (-1108,96%) SOLL 2: D Sectors IID Oxice [%] 24,284 (-115,10%) SOLS SM: Inter Executed [%] 8,10 (-105,86%) SOLL 2: D Sectors IID Oxice [%] 12,10 (-1																
SOL SM Breakdown SOL Memory Breakdown SOL, Microsov Active [N] 20.05 (+ 1000/15) SOL 2: Xbar2ns Cycles Active [N] 56.82 (+ 12.42) SOL SM: Inst Executed [%] 20.06 (+ 1000/15) SOL 2: The preparate [N] 56.82 (+ 12.42) SOL SM: Inst Executed [%] 10.83 (+ 1068091) SOL 2: The preparate [N] 30.81 (+ 1000/15) SOL SM: Inst Executed [%] 10.83 (+ 1068091) SOL 1: M LiterAcbar Rec Cycles Active [N] 30.81 (+ 1000/15) SOL SM: Inst Executed [%] 10.30 (+ 107.8991) SOL 1: Lus Windback Active [N] 30.10 (+ 100.8991) SOL SM: Inst Executed Pipe Lus [N] 10.30 (+ 107.8991) SOL 1: Lus Windback Active [N] 31.52 (+ 22.44) SOL SM: Inst Executed Pipe Lus [N] 8.10 (+ 101.8991) SOL 1: Lus Windback Active [N] 22.46 (+ 101.8991) SOL SM: Inst Executed Pipe Lus [N] 2.72 (+ 1068091) SOL L1: Lus Windback Active [N] 22.46 (+ 101.991) SOL SM: Inst Executed Pipe Lus [N] 2.72 (+ 1068091) SOL L1: Lus Windback Active [N] 22.46 (+ 101.991) SOL SM: Inst Executed Pipe Lus [N] 2.72 (+ 1068091) SOL L1: Lus Windback Active [N] 22.46 (+ 101.991) SOL SM: Inst Executed Pipe Lus [N] 2.72 (+ 1068091) SOL	0 100)	20.0	30.0		40.0	50.0	60.0	0	70.0	80.0	2	90.0	1	100	
SOL SM: Instaue Active [M] 20.01 (+160.09%) SOL L2: Xbar2its Cycles Active [M] 65.82 (+124.4) SOL SM: Inst Executed [Pi] 20.46 (+160.05%) SOL L2: Tog Requests [M] 46.67 (+178.4) SOL SM: Inst Executed [Pi] 20.46 (+160.05%) SOL L2: Xbar2its Cycles Active [M] 46.67 (+178.4) SOL SM: Inst Executed [Pi] 13.83 (+165.65%) SOL L2: Tog Requests [M] 36.86 (+170.4) SOL SM: Inst Executed [Pie] 13.83 (+165.65%) SOL L2: Tog Requests [M] 38.80 (+171.7) SOL SM: Inst Executed [Pie] La (M) 13.80 (+165.65%) SOL L2: Tog Requests [M] 38.80 (+171.7) SOL SM: Inst Executed [Pie] La (M) 13.03 (+165.65%) SOL L2: La Wavefronts [M] 34.80 (+174.7) SOL SM: Inst Executed [Pie] Cha (Pie) (M) 13.7 (+161.23%) SOL L2: La Wavefronts [M] 34.82 (+34.45%) SOL SM: Inst Executed [Pie Cha (Pie) (M) 13.13 (+165.65%) SOL L2: D Sectors [M] 22.64 (+151.45%) SOL SM: Inst Executed [Pie Cha (Pie) (M) 7.22 (+165.65%) SOL L1: La Wavefronts [M] 10.80 (+172.75%) SOL SM: Inst Executed [Pie Cha (Pie) (M) 7.23 (+165.65%) SOL L1: La Wavefronts [M] 10.80 (+172.75%) SOL SM: Inst Executed [Pie La (Pie) (M) <td>,0 10,0</td> <td>)</td> <td>20,0</td> <td>30,0</td> <td></td> <td></td> <td></td> <td> / -</td> <td>0</td> <td>70,0</td> <td>80,0</td> <td>D</td> <td>90,0</td> <td>1</td> <td>100</td>	,0 10,0)	20,0	30,0				/ -	0	70,0	80,0	D	90,0	1	100	
301. SM: Inst Executed (%) 20.46 (+16010%) SOL 21: Trap Requests (%) 452 (*178) 303. SM: Pipe Struct Cycles Active (%) 18.86 (+16010%) SOL 21: Trap Requests (%) 3061 (+107) 303. SM: Pipe Struct Cycles Active (%) 18.86 (+16050%) SOL L2: T Stertors (%) 3061 (+107) 303. SM: Pipe Struct Cycles Active (%) 18.86 (+16050%) SOL L2: T Stertors (%) 3061 (+107) 303. SM: Pipe Struct Cycles Active (%) 10.80 (+127)/6500 SOL L2: T Stertors (%) 38.80 (+71) 303. SM: Pipe Struct Cycles Active (%) 10.37 (+16123%) SOL DUP Dum Throughput (%) 38.25 (-42) 303. SM: And Pipe Struct Pipe Cuc Pied Or Any (%) 8.13 (+71.89%) SOL L1: Law Writeback Active (%) 24.84 (+15) 303. SM: And Pipe Struct Cycles Active (%) 8.10 (+105.86%) SOL L2: D Sectors (%) 22.46 (+15) 303. SM: And Pipe Fina Cycles Active (%) 8.10 (+105.86%) SOL L2: D Sectors (%) 12.10 (+12.16 (\0 10,0							/ -	-	,-			90,0	1	100	
D3: MP per Shared Cycles Active [N] 10.36 (+166.09%) SOL L1: M. L16x2Abar Reg Cycles Active [N] 30.61 (+107.216.00%) D3: MP per Shared Cycles Active [N] 10.36 (+166.09%) SOL L2: D settors [N] 38.61 (+107.216.00%) D3: MP to End Cycles Active [N] 10.36 (+112.29%) SOL L2: D bas Pipe Lux Wavefronts [N] 38.10 (+71.216.00%) D3: MP to End Cycles Active [N] 10.37 (+112.23%) SOL L2: D bas Pipe Lux Wavefronts [N] 34.25 (+42.26%) D3: MP to End Cycles Active [N] 8.10 (+112.29%) SOL L2: D bas Pipe and Cycles Active [N] 24.83 (+63.26%) D3: MM to End Active [N] 8.10 (+105.29%) SOL L2: D Sectors [N] 22.84 (+115.26%) D3: MM to End Active [N] 7.25 (+165.64%) SOL L2: D Sectors [N] 11.21 (+12.26%) D3: MM to End Active [N] 7.25 (+165.64%) SOL L2: D Sectors [N] 10.86 (+10.16%) D3: MM to End Active [N] 7.25 (+165.64%) SOL L2: D Sectors [N] 10.82 (+21.26%) D3: MM to End Active [N] 7.25 (+165.64%) SOL L2: D Sectors [N] 10.82 (+21.26%) D3: MM to End Active [N] 7.25 (+165.64%) SOL L2: D Sectors [N] 6.83 (+21.26%) D3: MM to End Active Dipe Active [N] 7.25 (+165),O 10,C							/ -	-	,-			90,0	1	10	
0LSM: Page Fipt6 Cycles Active [15] 18.36 (+1656.09%) SOL L2: T Sectors [15] 38.00 (+72) 0LSM: Inst Executed Pipe Lau [15] 10.80 (+127.96%) SOL L2: T Sectors [15] 34.00 (+76) 0LSM: Inst Executed Pipe Lau [15] 10.80 (+127.96%) SOL L2: T Sectors [15] 34.10 (+76) 0LSM: Pipe ALV (2564 Active [15] 10.37 (+161.29%) SOL L2: Data Pipe Lau Wavefronts [15] 34.10 (+76) 0LSM: Pipe ALV (2564 Active [15] 0.54 (+161.49%) SOL L2: D Sectors [16] 24.26 (+165) 0LSM: Morg Nether Cycles Active [15] 8.10 (+165.96%) SOL L2: D Sectors [16] 22.64 (+116) 0LSM: Morg Nether Cycles Active [15] 8.10 (+165.96%) SOL L2: D Sectors [16] 22.64 (+116) 0LSM: Morg Nether Cycles Active [15] 7.23 (+165.86%) SOL L2: D Sectors [16] 12.89 (+12.19) 0LSM: Morg Nether Cycles Active [15] 7.23 (+165.86%) SOL L2: L162.20% (+16.10%) 10.88 (+12.19%) 10.88 (+12.19%) 10.89 (+12.19%) 0LSM: Morg Nether Cycles Active [16] 6.23 (+12.19%) SOL L2: L162.20% (+10.10%) 6.33 (+2.53) 10.83 (+12.19%) 10.83 (+12.19%) 10.83 (+12.19%) 10.83 (+12.19%) 10.83 (+12.19%) 10.83 (+12.19%) 10.83					20	5	peed Of Ligh	t [%]	sc	L Memory						
0.15.M: trait Securited Pape Lux [Vs] 10.80 (+122/96%) SOL L1: bata Pape Lux Wavefronts [Vs] 34.10 (*76. 0.15.M: Pape Autry (Sec Active (N) 10.37 (*1612336) SOL L2: D2: D2: D2: D2: D2: D2: D2: D2: D2: D	sue Active [%]					.51 (+160.09%	() SOL L2:	t [%] Xbar2lts Cycle	SC es Active [%]	L Memory			55.82	(+124.58	%]	
0.13M: Pipe Alu Cycles Active [5] 10.327 (+161.235) SOL OPU: Deam Throughput [6] 31.52 (+22.) 0.13M: Bipe Alu Cycles Active [5] 0.13M: Pipe Alu Cycles Active [5] 24.83 (+31.23) 0.13M: Bipe Alu Cycles Active [5] 8.10 (+161.99%) SOL L1: Use Writeback Active [5] 24.84 (+161.23) 0.13M: Mo2P (Writeback Active [5] 2.81 (+161.99%) SOL L2: D Sectors [10] 22.64 (+116.23) 0.13M: Mo2P (Writeback Active [5] 2.753 (+165.64%) SOL L2: D Sectors [10] 22.64 (+116.23) 0.13M: Mo2P (Writeback Active [5] 7.253 (+165.64%) SOL L2: D Sectors [10] 22.64 (+116.23) 0.13M: Mo2P (Writeback Active [5] 7.252 (+165.64%) SOL L2: D Sectors [10] 63.82 (+21.23) 0.13M: Mo1P (Writeback Active [5] 7.252 (+165.64%) SOL L2: L12:200(+10) 10.82 (+21.23) 0.13M: Mo1P (Writeback Active [5] 7.25 (+165.64%) SOL L2: L12:200(+10) 10.82 (+21.23) 0.13M: Mo1P (Writeback Active [5] 7.25 (+165.64%) SOL L2: L12:200(+10) 10.82 (+21.23) 0.13M: Mo1P (Writeback Active [5] 7.22 (+165.80%) SOL L2: L12:200(+10) 10.82 (+21.23) 0.13M: Mo1P (Writeback Active [5] 7.21 (+165.80%) SOL L2: Data Bank Wr	sue Active [%]	SOL SM			20.	51 (+160.09% 46 (+160.10%	() SOL L2: SOL L2:	t [%] Xbar2its Cycle T Tag Request	SC es Active [%] ts [%]	L Memory			55.82	(+124.58 (+178.82	%]	
0.15M: https://www.sci.uk/sc	isue Active [%] ist Executed [%] ipe Shared Cycles A	SOL SM			20. 18.	51 (+160.09% 46 (+160.10% 36 (+165.69%	(i) SOL L2: (i) SOL L2: (i) SOL L2: (i) SOL L1:	t [%] Xbar2lts Cycle T Tag Request M L1tex2xbar R	sC es Active [%] ts [%] Req Cycles A	L Memory			55.82 45.27 39.61	(+124.58 (+178.82 (+107.93	%) %) %)	
00. SM: Inst Sexusted Pipe CDe Ped On Any [%] 8.13 (+7.98%) SOLL1: Law Writeback Active [%] 24.83 (+63.53) 0.05. Mic 207 Writes Active [%] 8.10 (+105.94%) SOLL2: D Sectors [#] Onice [%] 22.64 (+115.53) 0.05. Mic Any Private Cycles Active [%] 8.10 (+105.94%) SOLL2: D Sectors [#] Onice [%] 12.10 [= (1.2.12) 0.05. Mic Any Private Cycles Active [%] 8.10 (+105.96%) SOLL2: D Sectors [#] Onice [%] 10.80 (+127.53) 0.05. Mic Any Write Cycles Active [%] 7.22 (+165.86%) SOLL1: Law Requests [%] 0.892 (-21.53) 0.05. Mic Inst Issued [%] 5.55 (+122.69%) SOLL2: Liszbare Cycles Active [%] 6.892 (-21.53) 0.05. Mic Inst Issued [%] 4.59 (+166.56%) SOLL1: Law Requests [%] 6.892 (-21.53) 0.05. Mic Inst Executed Pipe Xu [%] 4.59 (+166.55%) SOLL1: Law Requests [%] 6.892 (-21.53) 0.05. Mic Inst Executed Pipe Xu [%] 4.59 (+166.55%) SOLL1: Law Requests [%] 6.892 (-21.53) 0.05. Mic Inst Executed Pipe Xu [%] 1.10 (+165.55%) SOLL1: Law Read Reseture [%] 6.892 (-21.53) 0.05. Mic Inst Executed Pipe Xu [%] 1.10 (+165.55%) SOLL1: Law Read Reseture [%] 6.90 (-21.54) 0.05. Mic Inst Exe	isue Active [%] ist Executed [%] ipe Shared Cycles A ipe Fp64 Cycles Act	SOL SM ctive [%] ive [%]			20. 18. 18.	51 (+160.09% 46 (+160.10% 36 (+165.69% 36 (+165.69%	() SOL L2: () SOL L2: () SOL L2: () SOL L2: () SOL L2:	t [%] Xbar2lts Cycle T Tag Request M Litex2xbar R T Sectors [%]	SC es Active [%] ts [%] Req Cycles A	L Memory			55.82 45.27 39.61 38.80	(+124.58' (+178.82' (+107.93') (+71.23'	%) %) %)	
00, SM: Mo27 Writeback Active [%] 8.10 (+101.97%) SOL L2: D Sectors [%] 22.64 (+15) 00, SM: Mo27 Writeback Active [%] 8.10 (+105.96%) SOL L2: D Sectors Fill Device [%] 12.19 (+12.19) 0.5 M/: Mo P4 mice Cycles Active [%] 7.85 (+166.66%) SOL L1: Lisuin Requests [%] 10.80 (+127.37) 0.5 M/: Mo P4 mice Cycles Active [%] 7.23 (+166.66%) SOL L2: LisZback Cycles Active [%] 8.82 (+12.19) 0.5 M/: Mo P4 mice Mice [%] 5.55 (+12.2960) SOL L2: LisZback Cycles Active [%] 6.82 (+12.19) 0.5 M/: Mo P4 mice Mice [%] 5.55 (+12.2960) SOL L2: LisZback Sectors [%] 6.93 (+25.19) 0.5 M/: Inst Executed Pipe Xin [%] 4.59 (+166.69%) SOL L1: Data Bank Writes [%] 3.09 (+7.27) 0.5 M: Inst Executed Pipe Aut [%] 1.96 (+165.59%) SOL L1: Data Bank Writes [%] 1.96 (+23.81) 0.5 M: Inst Executed Pipe Aut [%] 1.96 (+23.83) SOL L1: Texah Bank Writes [%] 0.00 (+7.26.84) 0.5 M: Inst Executed Pipe Aut [%] 1.96 (+35.53%) SOL L1: Texah Bank Writes [%] 0.00 (+23.84)	isue Active [%] ist Executed [%] ipe Shared Cycles A ipe Fp64 Cycles Act ist Executed Pipe Ls	SOL SM ctive [%] cive [%] cu [%]			20. 18. 18. 10.	51 (+160.09% 46 (+160.10% 36 (+165.69% 36 (+165.69% 80 (+127.96%	(i) SOL L2: (i) SOL L2: (i) SOL L2: (i) SOL L1: (i) SOL L2: (i) SOL L1: (i) SOL L2: (i) SOL L1:	t [%] Xbar2lts Cycle T Tag Request M L1tex2xbar R T Sectors [%] Data Pipe Lsu V	SC es Active [%] ts [%] Req Cycles A Wavefronts	L Memory			55.82 45.27 39.61 38.80 34.10	(+124.58 (+178.82 (+107.93) (+71.23) (+76.03	%) %) %) %)	
LO, MK. Moh P, Read Cycles Active (%) 8.10 (-105.89%) SOLL 2: D Sectors FID Drives (%) 12.16 (-12. LO, SM. Moh P, Ward Cycles Active (%) Z55 (-165.66%) SOLL 1: Lisuin Requests (%) 10.80 (-127. LO, SM. Moh P, Ward Cycles Active (%) Z52 (-165.66%) SOLL 1: Lisuin Requests (%) 68.82 (-21. LO, SM. Moh P, Ward Cycles Active (%) Z52 (-165.66%) SOLL 1: Lisuin Requests (%) 68.92 (-25. LO, SM. Moh P, Ward Cycles Active (%) 55.5 (-122.69%) SOLL 1: Lisuin Requests (%) 63.99 (-25. LO, SM. Moh T, Ward M, Mark D, Mark	isue Active [%] ist Executed [%] ipe Shared Cycles A ipe Fp64 Cycles Act ist Executed Pipe Ls ipe Alu Cycles Active	SOL SM .ctive [%] .ive [%] .eu [%] e [%]	1 Breakdo		20. 18. 18. 10. 10.	51 (+160.09% 46 (+160.10% 36 (+165.69% 36 (+165.69% 80 (+127.96% 37 (+161.23%	SOL L2: i) SOL L2: ii) SOL L2: iii) SOL L1: iii) SOL L2: iii) SOL L2: iiii) SOL L2: iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	t [%] Xbar2lts Cycle T Tag Request M L1tex2xbar R T Sectors [%] Data Pipe Lsu V U: Dram Throug	SC es Active [%] ts [%] Req Cycles A Wavefronts ighput [%]	L Memory			55.82 45.27 39.61 38.80 34.10 31.52	(+124.58 (+178.82 (+107.93) (+71.23) (+76.03 2 (-42.82	%) %) %) %) %)	
JOL SM: MIN PQ Write Cycles Active [%] 753 (±165.64%) SOL L1: Lusin Requests [%] 10.80 (±27) JOL SM: Pipe Fma Cycles Active [%] 7.22 (±165.66%) SOL L2: Lts.2bar Requests [%] 8.82 (±21) JOL SM: Min Issi secured [%] 5.55 (±12.56%) SOL L2: Lts.2bar Requests [%] 6.83 (±21) JOL SM: Inst Securited Pipe Xu [%] 4.59 (±165.66%) SOL L1: Total Bark Reads [%] 6.30 (±72) JOL SM: Inst Securited Pipe Xu [%] 4.59 (±165.66%) SOL L1: Total Bark Reads [%] 3.00 (±7) JOL SM: Inst Securited Pipe Xu [%] 1.95 (±12.55%) SOL L1: Total Bark Reads [%] 3.09 (±7) JOL SM: Inst Securited Pipe Aul [%] 1.18 (±165.55%) SOL L1: Total Bark Reads [%] 3.09 (±7) JOL SM: Inst Securited Pipe Aul [%] 1.18 (±165.55%) SOL L1: Total Bark Reads [%] 3.09 (±7)	isue Active [%] ist Executed [%] ipe Shared Cycles Ad ipe Ep64 Cycles Active ist Executed Pipe Ls ipe Alu Cycles Active ist Executed Pipe Cl	SOL SM .ctive [%] .tive [%] .eu [%] .e [%] .bu Pred On An	1 Breakdo		20. 18. 18. 10. 10. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8.	51 (+160.09% 46 (+160.10% 36 (+165.69% 36 (+165.69% 80 (+127.96% 37 (+161.23% 313 (+71.98%	SOL L2: i) SOL L2: ii) SOL L2: iii) SOL L2: iii) SOL L1: iii) SOL L2: iii) SOL L2: iii) SOL L2: iiii) SOL L2: iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	t [%] Xbar2lts Cycle T Tag Request M Litex2xbar R T Sectors [%] Data Pipe Lsu V U: Dram Throug Lsu Writeback	SC es Active [%] ts [%] Req Cycles A Wavefronts ughput [%] Active [%]	L Memory			55.82 45.27 39.61 38.8(34.1(31.52 24.83	(+124.58 (+178.82 (+107.93) (+71.23) (+76.03 2 (-42.82 (+63.67	%) %) %) %) %) %)	
JOB, Mr. Dippe Franc Cycles Active [N] 7.22 (+165.80%) SOLL 2: Lts2barb Cycles Active [N] 8.82 (-21. JOB, SM. Mol hant Issued [N] 5.55 (+12.200%) SOLL 1: M Xbar2/Tites Read Sectors [N] 6.39 (-25. JOB, SM. Mol hant Issued [N] 5.55 (+12.200%) SOLL 1: M Xbar2/Tites Read Sectors [N] 6.39 (-25. JOB, Kin Inst Executed Pipe Xu [N] 4.59 (+165.50%) SOLL 1: Data Bank Reads [N] 3.00 (-77. JOB, Kin Inst Executed Pipe Aul [N] 1.19 (+165.55%) SOLL 1: Data Bank Reads [N] 1.39 (-165.5%) JOB, Kin Inst Executed Pipe Aul [N] 1.119 (+165.5%) SOLL 1: Data Bank Reads [N] 1.39 (-17.5%)	isue Active [%] ist Executed [%] ipe Shared Cycles A ipe Fp64 Cycles Activ ist Executed Pipe Lis it Executed Pipe Ct tio2rf Writeback Activ	SOL SM .ctive [%] .ive [%] e [%] bu Pred On An ive [%]	1 Breakdo		20. 18. 18. 10. 10 8 8	51 (+160.09% 46 (+160.10% 36 (+165.69% 36 (+165.69% 36 (+165.69% 80 (+127.96% 37 (+161.23% 3.13 (+71.98% 1.10 (+161.91%	SOLL2:) SOLL1:) SOLL2:) SOLL2:) SOLL2:) SOLL2:) SOLL3:) SOLL4:) SOLL4:) SOLL1:) SOLL2:	t [%] Xbar2lts Cycle T Tag Request M Litex2xbar R T Sectors [%] Data Pipe Lsu V U: Dram Throug Lsu Writeback D Sectors [%]	SC es Active [%] ts [%] Req Cycles A Wavefronts ighput [%] Active [%]	L Memory			55.82 45.27 39.61 38.80 34.10 31.52 24.83 22.64	(+124.58' (+178.82' (+107.93') (+71.23') (+76.03' 2 (-42.82' (+63.67' 4 (+115.71'	%) %) %) %) %) %)	
XDL SM: Milo inst Issued [%] 5.55 (+123.69%) SOL L1: M Xbar2/Itex Read Sectors [%] 6.39 (-25. OL SM: Inst Executed Pipe Xu [%] 4.59 (+66.69%) SOL L1: M Xbar2/Itex Read Sectors [%] 3.00 (+7. OL SM: Inst Executed Pipe Xu [%] 4.59 (+66.69%) SOL L1: M Xbar2/Itex Read Sectors [%] 3.00 (+7. OL SM: Inst Executed Pipe Inflorm [%] 1.29 SOL L1: To ata Bank Winter [%] 1.95 (+23.5 OL SM: Inst Executed Pipe Adu [%] 1.18 (+165.55%) SOL L1: Texin Sm2tex Req Cycles Active [%] 0.00 (+258.4	isue Active [%] ist Executed [%] ipe Shared Cycles A ipe Fp64 Cycles Active St Executed Pipe Ls ipe Alu Cycles Active st Executed Pipe Cl i02rf Writeback Acti li0 Pq Read Cycles A	SOL SM	1 Breakdo		20. 18. 18. 10. 10. 10 8. 8. 8.	51 (+160.09% 46 (+160.10% 36 (+165.69% 36 (+165.69% 36 (+127.96% 37 (+161.23% 313 (+71.98%) 10 (+161.91% 10 (+105.96%)	SOLL2:) SOLL2:) SOLL2:) SOLL2:) SOLL1:) SOLL2:) SOLL2:	t [%] Xbar2lts Cycle T Tag Request M Litex2xbar R T Sectors [%] Data Pipe Lsu V U: Dram Throug Lsu Writeback. D Sectors [%] D Sectors Fill [SC es Active [%] ts [%] Req Cycles A Wavefronts ighput [%] Active [%] Device [%]	L Memory			55.82 45.27 39.61 38.8(34.10 31.52 24.83 22.64 12.11	(+124.58' (+178.82' (+107.93') (+71.23') (+76.03' 2 (-42.82' 1 (+63.67' 4 (+115.71' 9 (-12.29'	%) %) %) %) %) %)	
0L.Sk: trat Securited Pipe Xu (%) 4.59 (+166.69%) SOL L1: bata Bank Reads (%) 3.09 (-77) OL Sk: trat Securited Pipe Xu (%) 1.29 SOL L1: bata Bank Writes (%) 1.95 (+23) OL Sk: trat Securited Pipe Adu (%) 1.18 (+165.55%) SOL L1: Toxin Sm2tzer Ref Cycles Active (%) 0.00 (-258)	isue Active [%] ist Executed [%] ipe Shared Cycles A ipe Fp64 Cycles Activ ist Executed Pipe L iio2rf Writeback Activ iio Pq Read Cycles A iio Pq Write Cycles A	SOL SM .ctive [%] .tv [%] a [%] bu Pred On An ive [%] .ctive [%] .ctive [%]	1 Breakdo		20. 18. 18. 10. 10. 8. 8. 8. 7.	51 (+160.09% 46 (+160.10% 36 (+165.69% 80 (+127.96% 80 (+127.96% 37 (+161.23% 3.13 (+71.98% 1.10 (+161.91% 10 (+105.96% 53 (+165.64%	Append Of Light (a) SOLL2: (b) SOLL2: (c) SOLL2: (c) SOLL1: (c) SOLL2: (c) SOLL2: (c) SOLL2: (c) SOLL2: (c) SOLL2: (c) SOLL1: (c) SOLL2:	t [%] Xbar2lts Cycle T Tag Request M Litex2xbar R T Sectors [%] Data Pipe Lsu V J: Dram Throug Lsu Writeback D Sectors [%] D Sectors Fill [Lsuin Requests	SC es Active [%] ts [%] Req Cycles A Wavefronts ighput [%] Active [%] Device [%] s [%]	L Memory			55.82 45.27 39.61 38.8(34.10 31.52 24.83 22.64 12.11 10.80	(+124.58' (+178.82' (+107.93') (+71.23') (+71.23') (+76.03' 2 (-42.82' (+63.67' 4 (+115.71' 9 (-12.29' (+127.96'	%) %) %) %) %) %) %) %)	
OL SM: Inst Executed Pipe Uniform [½] 1.29 SOL L1: Data Bank Writes [½] 1.95 (+23.3 OL SM: Inst Executed Pipe Adu [½] 1.18 (+165.53%) SOL L1: Texin Sm2tex Req Cycles Active [½] 0.00 (+258.4	isue Active [%] ist Executed [%] ipe Shared Cycles Act ist Executed Pipe Ls ipe Alu Cycles Act ist Executed Pipe Cl lio2rf Writeback Act ito Pa Read Cycles A ipe Fma Cycles Act	SOL SM .ctive [%] .tv [%] a [%] bu Pred On An ive [%] .ctive [%] .ctive [%]	1 Breakdo		20. 18. 10. 10 8 8 8. 7. 7. 7.	51 (+160.09) 46 (+160.10) 36 (+165.69) 36 (+165.69) 80 (+127.96) 37 (+161.23) 313 (+71.98) 10 (+1161.91) 10 (+105.96) 53 (+165.64) 22 (+165.80)	Append Of Light (a) SOLL2: (b) SOLL2: (c) SOLL2: (c) SOLL2: (c) SOLL1: (c) SOLL2: (c) SOLL2: (c) SOLL2: (c) SOLL1: (c) SOLL2: (c) SOLL2: (c) SOLL2: (c) SOLL2: (c) SOLL2: (c) SOLL2:	t [%] Xbar2lts Cycle T Tag Request M Litex2xbar R T Sectors [%] Data Pipe Lsu U U: Dram Throug Lsu Writeback D Sectors [%] D Sectors Fill Esuin Requests Lts2xbar Cycle	SC es Active [%] ts [%] Req Cycles A Wavefronts ghput [%] Active [%] Device [%] s [%] es Active [%]	PL Memory			55.82 45.27 39.61 38.80 34.10 31.52 24.83 22.84 12.11 10.80 8.83	(+124.58' (+178.82' (+107.93') (+71.23') (+71.23') (+76.03' 2 (-42.82' (+63.67' 4 (+115.71' 9 (-12.29' (+127.96' 2 (-21.48'	%) %) %) %) %) %) %) %) %)	
KOL SM: Inst Executed Pipe Adu [%] 1.18 (+165.53%) SOL L1: Texin Sm2tex Req Cycles Active [%] 0.00 (+258.63%)	isue Active [%] ist Executed [%] ipe Fp64 Cycles A ipe Fp64 Cycles Active ist Executed Pipe Ls ipe Au Cycles Active ist Executed Pipe Ct lio Pa Read Cycles A tio Pa Write Cycles A tio Pa Write Cycles A ipe Fma Cycles Activ	SOL SM ctive [%] ive [%] au [%] au [%] ou Pred On An ive [%] ictive [%] ictive [%] ictive [%]	1 Breakdo		20. 18. 10. 10 8 8. 7. 7. 7. 5.	51 (+160.09) 46 (+160.09) 36 (+165.69) 36 (+165.69) 80 (+127.96) 81 (+127.96) 81 (+127.96) 81 (+127.96) 81 (+127.96) 81 (+127.96) 53 (+165.64) 55 (+123.69)	Solution a) Solution b) Solution c) Solution	t [%] Xbar2lts Cycle T Tag Request M Litex2xbar R T Sectors [%] Data Pipe Lsu U U: Dram Throug Lsu Writeback. D Sectors [%] D Sectors [%] D Sectors Fill [Lsuin Requests Lts2xbar Cycle M Xbar2litex R	SC es Active [%] ts [%] Req Cycles A Wavefronts gghput [%] Active [%] Device [%] s [%] Read Sectors	PL Memory			55.82 45.27 39.61 38.86 34.10 31.52 24.83 22.64 12.11 10.80 8.88 6.39	(+124.58' (+178.82' (+107.93') (+71.23') (+76.03' 2 (-42.82' (+63.67' 4 (+115.71' 9 (-12.29' (+127.96' 2 (-21.48' 9 (-25.58'	%) %) %) %) %) %) %) %) %) %)	
	isue Active [%] inst Executed [%] ipe Shared Cycles At pe Fp64 Cycles At St Executed Pipe Ls ipe Alu Cycles Active inst Executed Pipe Cl ioo2rd Writeback Acti ilio Pa Write Cycles A ipe Fma Cycles Acti- filo Inst Issued [%]	SOL SM	1 Breakdo		20. 18. 10. 10 8 8. 7. 7. 7. 5.	\$ 51 (+160.09) 46 (+166.09) 36 (+165.69) 36 (+165.69) 37 (+161.23) 37 (+161.23) 13 (+71.98) 10 (+161.91) 10 (+165.96) 55 (+165.64) 22 (+165.64) 55 (+165.64) 55 (+165.69) 55 (+165.69)	Soll Soll a) Soll Soll b) Soll Soll	t [%] Xbar2lts Cycle T Tag Request M Litex2xbar R T Sectors [%] Data Pipe Lsu \ U: Dram Throug Lsu Writeback. D Sectors [%] D Sectors [%] Lsuin Requests Lts2xbar Cycle M Xbar2ltex R Data Bank Read	SC es Active (%) ts (%) Req Cycles A Wavefronts i gghput (%) Active (%) Bevice (%) s (%) es Active (%) Read Sectors tds (%)	PL Memory			55.82 45.27 39.61 38.88 34.10 31.52 24.83 22.64 12.11 10.80 8.88 8.6.33 3.0	(+124.58' (+178.82' (+107.93') (+71.23') (+76.03' 2 (-42.82' (+63.67' 4 (+115.71' 9 (-12.29' 2 (-21.48' 9 (-25.58' 9 (+77.11'	%) %) %) %) %) %) %) %) %) %) %) %) %) %	
SUC IDC: Request Cycles Active [76] 0.00 (+258.0 (+258.0 0.00 (+258.0 0.00 (+258.0 0.00 (+258.0 0.00 (+258.0 0.00 (+258.0 0.00 (+258.0 0.00 (+258.0 0.00 (+258.00 (+258.0 0.00 (+258.0 0.00 (+258.00	isue Active [%] inst Executed [%] ipe Shared Cycles A ipe Fp64 Cycles Active ist Executed Pipe L ipe Alu Cycles Active ist Executed Pipe C ibo27 Writeback Acti ito Pq Read Cycles Active file Pf Read Cycles Active file Pf Read Cycles Active file Fina Cycles Active to Inst Issued [%] ist Executed Pipe U	SOL SM ctive [%] ive [%] tu [%] bu Pred On An ive [%] bu Ure(%) kctive [%] kctive [%] j [%] j [%]	1 Breakdo		20. 18. 10. 10. 10. 5. 8. 8. 7. 7. 7. 5. 4.	\$.51 (+160.09% 46 (+160.10% 36 (+165.69% 36 (+165.69% 37 (+161.23% 1.3 (+71.98% 1.10 (+161.91%) 1.0 (+165.64% 52 (+165.64%) 55 (+165.69%) 55 (+165.69%) 1.2	i) SOL L2: i) SOL L2: i) SOL L1: i) SOL L2: i) SOL L2: i) SOL L1: i) SOL L2: i) SOL L1: i) SOL L1: i) SOL L1: i) SOL L1:	t [%] Xbar2lts Cycle T Tag Request M Litex2xbar R T Sectors [%] Data Pipe Lsu V U: Dram Throug Lsu Writeback. D Sectors Fill D Sectors Fill D Sectors Fill C Lsuin Requests Lts2xbar Cycle M Xbar2litex R Data Bank Read Data Bank Writ	SC es Active [%] ts [%] Req Cycles A Wavefronts ghput [%] Active [%] Device [%] s [%] es Active [%] Read Sectors (%) ftes [%]	Ctive [%]			55.82 45.27 39.61 38.80 34.10 31.55 24.83 22.84 12.11 10.80 8.83 6.39 3.00 1.95	(+124.58' (+178.82' (+107.93') (+716.33') (+716.33' 2 (-42.82' 1 (+63.67' 4 (+115.71' 9 (-12.29' (+127.96' 2 (-21.48' 9 (+77.11' (+23.96'	%] %] %] %] %] %] %] %] %] %] %]	
201 014 (-0.001) 001 (0.000) 001 (0.000) 000 (-0.000)	isue Active [%] ist Executed [%] ipe Shared Cycles Act ist Executed Pipe Ls ist Executed Pipe Ls ist Executed Pipe Cl fio2rf Writeback Act iso Pa Read Cycles A fio Pa Read Cycles A iso Pa Read Cycles A ist Executed Pipe X, ist Executed Pipe X, ist Executed Pipe X.	SOL SM	1 Breakdo		20. 18. 10. 10. 8. 8. 7. 7. 7. 5. 4.	\$ 51 (+160.09) 46 (+160.10) 36 (+165.69) 36 (+165.69) 80 (+127.96) 80 (+127.96) 80 (+127.96) 81 (+105.96) 53 (+165.64) 22 (+165.69) 55 (+123.69) 55 (+123.69) 55 (+123.69) 51 (+165.53) 12 18 (+165.53)	SOL L2: i) SOL L2: ii) SOL L2: iii) SOL L1: iii) SOL L2: iii) SOL L1:	t (%) Xbar2lts Cycle T Tag Request T Tag Request M LitezZhar R T Sectors (%) Data Pipe Lsu V U: Dram Throupus Lsu Writeback D Sectors (%) D Se	SC es Active [%] ts [%] Req Cycles A Wavefronts gighput [%] Active [%] es Active [%] es Active [%] Read Sectors ids [%] Read Sectors ids [%] Req Cycles A	Ctive [%]			55.82 45.27 39.61 38.80 34.10 31.55 22.64 12.11 10.80 8.83 6.33 3.00 1.95 0.000	(+124.58' (+178.82' (+107.93') (+71.23') (+71.23') (+72.03' 2 (-42.82' 1 (+63.67' 4 (+115.71' 9 (-12.29' (+127.96' 2 (-21.48' 9 (+27.11' i (+23.96' (+258.69'	%) %) %) %) %) %) %) %) %) %) %) %) %) %	
SOL SM: Inst Executed Pipe Tex [%] 0 (+0.00%) SOL L2: D Sectors Fill Sysmem [%] 0.00 (+i SOL SM: Inst Executed Pipe Ipa [%] 0 (+0.00%) SOL L1: D Sectors Fill Sysmem [%] 0.00 (+i	isue Active [%] ist Executed [%] ipe Shared Cycles Active ipe F064 Cycles Active ist Executed Pipe L flio2rf Writeback Active ist Executed Pipe Active ipe Fma Cycles Active flio Pa Write Cycles Active ipe Fma Cycles Active ist Executed Pipe Active ist Piecetories Pi	SOL SM ctive [%] ive [%] a [%] a [%] uu [%] uu [%] tctive [%] a [%] nform [%] du [%] e [%]	1 Breakdo		20. 18. 10. 10. 8. 8. 7. 7. 7. 5. 4.	\$ 51 (+160.09) 46 (+160.10) 36 (+165.69) 38 (+167.96) 37 (+161.23) 37 (+161.23) 37 (+161.23) 37 (+161.23) 38 (+179.96) 10 (+105.96) 53 (+165.64) 55 (+123.69) 59 (+165.69) 1.2 18 (+165.53)	ppeed of Light) SOL L2:) SOL L3:) SOL L4:) SOL L4:	t (%) Xbar2lts Cycle T Tag Request M Litex2bar7 Data Pipe Lsu u U: Dram Throug Lsu Writeback U: Dram Throug Lsu Writeback D Sectors [%] D Sectors [%] D Sectors [%] D Sectors Cycle M Xbar2Ites R Data Bank Rea Tasin Sm2tes R Fuel Sectors (%) Sectors (%) Data Sectors (%) Data Sectors (%) Data Sectors (%) Data Sectors (%) Sectors (%) Sectors (%) Sectors (%) Sectors (%) Data Sectors (%) Data Sectors (%) Data Sectors (%) Data Sectors (%) Data Sectors (%) Sectors (%) Data Sectors (%) D Sector	SC es Active [%] ts [%] Req Cycles A Wavefronts ghput [%] Active [%] es Active [%] Read Sectors kds [%] tes [%] Req Cycles A [%]	ctive [%]			55.82 45.27 39.61 38.86 34.10 31.55 24.83 22.64 12.11 10.80 8.83 6.39 3.0 1.95 0.00 0.000	(+124.58' (+178.82' (+178.82' (+107.93') (+71.23') (+71.23') (+76.03' (+72.82') (+72.84') (+127.96' (+127.96') (+21.48') (+25.58') (+22.96') (+22.96.69') (+258.69')	%) %) %) %) %) %) %) %) %) %) %) %) %) %	

Programming GPUs Advanced Topics

Advanced Topics

So much more interesting things to show!

- Optimize memory transfers to reduce overhead
- Optimize applications for GPU architecture
- Drop-in BLAS acceleration with NVBLAS (\$LD_PRELOAD)
- Tensor Cores for Deep Learning
- Libraries, Abstractions: Kokkos, Alpaka, Futhark, HIP, SYCL, C++AMP, C++ pSTL, ...
- Use multiple GPUs
 - On one node
 - Across many nodes \rightarrow MPI
-
- Some of that: Addressed at dedicated training courses



Using GPUs on JSC Systems

Compiling

- CUDA Module: module load CUDA/12
 - Compile: nvcc file.cu
 - Example cuBLAS:g++ file.cpp -I\$CUDA_HOME/include -L\$CUDA_HOME/lib64
 -lcublas -lcudart
- **OpenACC • Module:** module load NVHPC/24.3-CUDA-12
 - Compile: nvc++ -acc=gpu file.cpp



Compiling

- CUDA Module: module load CUDA/12
 - Compile: nvcc file.cu
 - Example cuBLAS:g++ file.cpp -I\$CUDA_HOME/include -L\$CUDA_HOME/lib64 -lcublas -lcudart
- **OpenACC Module:** module load NVHPC/24.3-CUDA-12
 - Compile: nvc++ -acc=gpu file.cpp
 - MPI CUDA-aware MPIs (with direct Device-Device transfers)

ParaStationMPI module load ParaStationMPI/5.9.2-1 MPI-settings/CUDA OpenMPI module load OpenMPI/4.1.5 MPI-settings/CUDA



Compiling

- CUDA Module: module load CUDA/12
 - Compile: nvcc file.cu
 - Example cuBLAS:g++ file.cpp -I\$CUDA_HOME/include -L\$CUDA_HOME/lib64 -lcublas -lcudart
- **OpenACC Module:** module load NVHPC/24.3-CUDA-12
 - Compile: nvc++ -acc=gpu file.cpp
 - MPI CUDA-aware MPIs (with direct Device-Device transfers)

ParaStationMPI module load ParaStationMPI/5.9.2-1 MPI-settings/CUDA OpenMPI module load OpenMPI/4.1.5 MPI-settings/CUDA

Containers Use containers via Apptainer (container group needed)

\$ apptainer pull tf.sif docker://nvcr.io/nvidia/tensorflow:20.12-tf1-py3
\$ srun -n 1 --pty apptainer exec --nv tf.sif python3 myscript.py



Running

Dedicated GPU partitions
 JUWELS

```
--partition=gpus 46 nodes (Job limits: \leq 1 d)
```

```
--partition=develgpus 10 nodes (Job limits: \leq2 h, \leq 2 nodes)
```

JUWELS Booster

```
--partition=booster 926 nodes
```

```
--partition=develbooster 10 nodes (Job limits: \leq 1 d, \leq 2 nodes)
```

JURECA DC

--partition=dc-gpu 192 nodes --partition=dc-gpu-devel 12 nodes

Optional

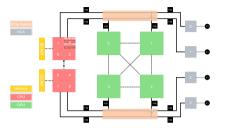
Resource Configuration --gres=gpu:4 Account --account myproject or jutil

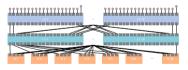
 \rightarrow See online documentation

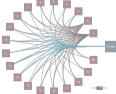


Running JUWELS Booster Topology

- JUWELS Booster: NPS-4 (in total: 8 NUMA Domains)
- Not all have GPU or HCA affinity!
- Network is structured into two levels: In-Cell and Inter-Cell (DragonFly+ network)







JÜLICH Forschungszentrum

→ Documentation: apps.fzjuelich.de/jsc/hps/juwels/

Example

- 16 tasks in total, running on 4 nodes
- Per node: 4 GPUs

```
#!/bin/bash -x
#SBATCH --nodes=4
#SBATCH --ntasks=16
#SBATCH --ntasks-per-node=4
#SBATCH --output=gpu-out.%j
#SBATCH --error=gpu-err.%j
#SBATCH --time=00:15:00
#SBATCH --account=training2310
```

```
#SBATCH --partition=dc-gpu
## SBATCH --partition=booster ## JWB
#SBATCH --gres=gpu:4
```

srun ./gpu-prog

Submit with sbatch run.sh



Conclusion

- GPUs provide highly-parallel computing power
- We have many devices installed at JSC, ready to be used!



- GPUs provide highly-parallel computing power
- We have many devices installed at JSC, ready to be used!
- Training courses by JSC next year
- See online documentation and sc@fz-juelich.de



- GPUs provide highly-parallel computing power
- We have many devices installed at JSC, ready to be used!
- Training courses by JSC next year
- See online documentation and sc@fz-juelich.de
- Further consultation via our lab: NVIDIA Application Lab in Jülich; contact me!



- GPUs provide highly-parallel computing power
- We have many devices installed at JSC, ready to be used!
- Training courses by JSC next year
- See online documentation and sc@fz-juelich.de
- Further consultation via our lab: NVIDIA Application Lab in Jülich; contact mel-





Appendix

Appendix Glossary References



Glossary I

- AMD Manufacturer of CPUs and GPUs. 42, 43, 44, 45, 46, 47, 80, 82
- Ampere GPU architecture from NVIDIA (announced 2019). 4, 5, 6
 - API A programmatic interface to software by well-defined functions. Short for application programming interface. 42, 43, 44, 45, 46, 47
 - CUDA Computing platform for GPUs from NVIDIA. Provides, among others, CUDA C/C++. 2, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 82
 - HIP GPU programming model by AMD to target their own and NVIDIA GPUs with one combined language. Short for Heterogeneous-compute Interface for Portability. 42, 43, 44, 45, 46, 47



Glossary II

- JSC Jülich Supercomputing Centre, the supercomputing institute of Forschungszentrum Jülich, Germany. 2, 74, 75, 76, 77, 81
- JURECA A multi-purpose supercomputer at JSC. 6
- JUWELS Jülich's new supercomputer, the successor of JUQUEEN. 3, 4, 5, 70
 - MPI The Message Passing Interface, a API definition for multi-node computing. 65, 67, 68, 69
 - NVIDIA US technology company creating GPUs. 3, 4, 5, 6, 16, 17, 18, 42, 43, 44, 45, 46, 47, 60, 74, 75, 76, 77, 80, 82
- OpenACC Directive-based programming, primarily for many-core machines. 36, 37, 38, 39, 40, 67, 68, 69



Glossary III

- OpenCL The Open Computing Language. Framework for writing code for heterogeneous architectures (CPU, GPU, DSP, FPGA). The alternative to CUDA. 42, 43, 44, 45, 46, 47
- OpenMP Directive-based programming, primarily for multi-threaded machines. 36, 37, 38, 39, 40
 - ROCm AMD software stack and platform to program AMD GPUs. Short for Radeon Open Compute (*Radeon* is the GPU product line of AMD). 42, 43, 44, 45, 46, 47
 - SAXPY Single-precision $A \times X + Y$. A simple code example of scaling a vector and adding an offset. 24, 57, 58
 - Tesla The GPU product line for general purpose computing computing of NVIDIA. 3



Glossary IV

- CPU Central Processing Unit. 3, 6, 11, 12, 13, 15, 16, 17, 18, 24, 42, 43, 44, 45, 46, 47, 80, 82
- GPU Graphics Processing Unit. 2, 3, 4, 5, 6, 8, 11, 12, 13, 14, 15, 16, 17, 18, 23, 27, 28, 29, 30, 31, 32, 35, 36, 37, 38, 41, 42, 43, 44, 45, 46, 47, 58, 59, 60, 64, 65, 66, 70, 71, 72, 74, 75, 76, 77, 80, 81, 82
- SIMD Single Instruction, Multiple Data. 15, 16, 17, 18
- SIMT Single Instruction, Multiple Threads. 14, 15, 16, 17, 18
 - SM Streaming Multiprocessor. 15, 16, 17, 18
- SMT Simultaneous Multithreading. 15, 16, 17, 18



References I

- [2] Karl Rupp. Pictures: CPU/GPU Performance Comparison. URL: https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardwarecharacteristics-over-time/ (pages 9, 10).
- [6] Wes Breazell. Picture: Wizard. URL: https://thenounproject.com/wes13/collection/its-a-wizards-world/ (pages 27-31).



References: Images, Graphics I

- [1] Forschungszentrum Jülich GmbH (Ralf-Uwe Limbach). JUWELS Booster.
- [3] Mark Lee. Picture: kawasaki ninja. URL: https://www.flickr.com/photos/pochacco20/39030210/ (pages 11, 12).
- [4] Shearings Holidays. Picture: Shearings coach 636. URL: https://www.flickr.com/photos/shearings/13583388025/ (pages 11, 12).
- [5] Nvidia Corporation. Pictures: Volta GPU. Volta Architecture Whitepaper. URL: https://images.nvidia.com/content/volta-architecture/pdf/Volta-Architecture-Whitepaper-v1.0.pdf.

