What's New in CUDA 5

Mark Ebersole

Computer Vision





The Soul of CUDA

The Platform for High Performance Parallel Computing

Accessible High Performance Enable Computing Ecosystem

GPU Accelerated Libraries "Drop-in" Acceleration for your Applications



Introducing CUDA 5

CUDA 5

Application Acceleration Made Easier

Dynamic Parallelism Spawn new parallel work from within GPU code on GK110

> GPU Object Linking Libraries and plug-ins for GPU code

New Nsight[™] Eclipse Edition Develop, Debug, and Optimize... All in one tool!

GPUDirect™ *RDMA between GPUs and PCIe devices*



FERMI 1 Work Queue

KEPLER 32 Concurrent Work Queues









Without Hyper-Q



Time

.....

With Hyper-Q



Time

.....

Dynamic Parallelism



What is CUDA Dynamic Parallelism?

The ability for any GPU thread to launch a parallel GPU kernel

- Dynamically
- Simultaneously
- Independently



Fermi: Only CPU can generate GPU work

Kepler: GPU can generate work for itself

Dynamic Work Generation



Higher Performance Lower Accuracy Lower Performance Higher Accuracy Target performance where accuracy is required

Familiar Syntax and Programming Model



Simpler Code: LU Example



LU decomposition (Kepler)



Mapping Compute to the Problem



Mapping Compute to the Problem



CUDA Dynamic Parallelism



CUDA 4: Whole-Program Compilation & Linking



CUDA 4 required single source file for a single kernel No linking external device code

CUDA 5: Separate Compilation & Linking



Separate compilation allows building independent object files CUDA 5 can link multiple object files into one program

CUDA 5: Separate Compilation & Linking



Can also combine object files into static libraries Link and externally call *device* code Facilitates code reuse, reduces compile time

CUDA 5: Separate Compilation & Linking

Enables closed-source device libraries to call user-defined device callback functions



NVIDIA[®] Nsight^{*} Eclipse Edition



Liter BCC++ D Debs

Global Store Efficience

Theoretical .

2,342 m

[256.1.1

256.1

©©©© Debug-findmax/src/findmax.cu-Cider File Edit Source Berlattor Navioate Search Run Protect Window Help													
[C+ 回 6 合 m 9+ 0+ 9+ 9 / / タ タ (0+ 0+ 0+ 0+ 0+ 0+ 0+ 0+ 0+ 0+ 0+ 0+ 0+ 0													
🏶 Debug B 💦 🖈 🐘 🗵 🗰 🗗 🕱 👁 🗷 🖷 🖛 🕱 📍 🖓 🖓	Information 13	% Breakpoints	1										
▼ E Findmax [C/C++ Application]	🔽 🐖 🔍 sm 2 war	rp 7											
* 🛐 cudaFindMax [0] [device: 0] (Suspended : Step)		Duraliza.	Desiden 0		Dre 4 dhana 📔								
CUDA Thread (0,0,0) Block (0,0,0)	· III (0) cudarmumax	Running	Device o	444(32,1,1),	,230,1,1)***								
cudaFindMax() at findmax.cu:1140x91f3a8	······································	Dunning	Joi 2	C findmax cut	111/0-01[310]								
 Ø CUDA Thread (1,0,0) Block (0,0,0) 	(224,0,0)	Dunning	Warp 7 Lane 0	Codmax.co.1	13 (0x911316)								
Block (0,0,0) [sm: 0] (256 Active Threads)	(225,0,0)	Bunoing	Warp 7 Lane 1	E findmax.cu.t	13 (0.911310)								
Block (1,0,0) [sm: 2] (256 Active Threads)	(220,0,0)	Running	warp / Lane 2	R findmax.cu.f	13 (0x911318)								
🕼 findmax.cu 🗱	E Outline III Disassembly III Registers II												
uint32_t nextElement;				5 45 D P +									
uint32_t i = firstElementIndex + threadsCount;													
<pre>for (: i < ARRAY SIZE: i += threadsCount) {</pre>			Name	T(0,0,0)B(0,0,0)	T(1,0,0)B(0,0,0)								
<pre>nextElement = array[i];</pre>			IIII RO	0	1								
<pre>if (nextElement > max) {</pre>			RI RI	16776272	16776272								
maxIndex = 1:		0	IIII R2	4935629	2024586								
)	₽		III R3	8192	8193								
}			ZUI R4	3149939	8115414								
threadMaxIdx[threadIdx.x] = maxIndex:			WW RS	4	4								
			222 R6	1048576	1048576								
	200 R7	4	4										
Console 22		III RS	32768	32772									
Endman [C/Cost Application] Endman		211 R9	0	0									
Running single-threaded host code	211 R10	8387951	16778240										
Max number is 0x800000 with index 2737098	202 R11	0	0										
Remains welki Absorded device ande	200 R12	1048576	1048576										
Running mittl-threaded device code			222 R13	0	0								
A.													





Automated CPU to GPU code refactoring Semantic highlighting of CUDA code

Integrated code samples & docs

Nsight Debugger

- Simultaneously debug of CPU and GPU
- Inspect variables across CUDA threads
- Use breakpoints & single-step debugging

Nsight Profiler

- ----

in Thread -151415062

Driver API

Profiling Overher

Deforce GTX 480

T MemCpy (HoC) T MemCpy (Date)

20.3% [4] Wclofs2int*, in

7 0.0% [4] VecEmptyLeokD

Analysis II III Details Cons

Quickly identifies performance issues

Low Global Memory Load Efficiency [9% avg. for kernels accounting for 75.6% of compute Low Global Memory Store Efficiency (21.3% ave. for kernels accounting for 73.9% of compute

- Integrated expert system
- Source line correlation

Available for Linux and Mac OS

NVIDIA[®] Nsight,^{*} Eclipse Edition



CUDA aware editor

- Integrated CUDA samples makes it quick and easy to get started
- Easily port CPU loops to CUDA kernels with automatic code refactoring
- Semantic highlighting of CUDA code makes it easy to differentiate GPU code from CPU code
- Generate code faster with CUDA aware auto code completion and inline help
- Hyperlink navigation enables faster code browsing
- Supports automatic makefile generation

NVIDIA[®] Nsight," Eclipse Edition

🧟 🗇 🛛 Debug - findmax/src/findmax.cu - Nsight												
File Edit Source Refactor Navigate Search Project Run Window Help												
	19 🗟 🚔 🗟 🕸 🗴 🖓 🛪 🖓 🗴 🖓 🖉 🖓 🗸 🖓 🗸 🖓 🗸 🖓	*⇔ ⇔•	->-					🖬 🏇 😪				
🕸 Debug 🕱 📃 🗣 Breakpo					ints 🕮 Registers 🛋 Modules 🜊 CUDA 🕱 💿 🖻 📑							
× % # # • 0 • • * 3												
Findmax [C/C++ Application]			[0] cudaFindMax Device 0			3	32 blocks of 32 are running					
🔻 🌮 cudaFindMax [0] [device: 0] (Suspended : Step)			▼ 🍥 (0,0,0)		SM 0 256 th		56 threads of 256 are r	unning 🖯				
CUDA Thread (0,0,0) Block (0,0,0)			(160,0,0)		Warp 5 Lane 0 🚺 findr		ndmax.cu:96 (0x9340b	8)				
= cudaFindMax() at findmax.cu:98 0x934168			@ (161,		Warp 5 Lane 1 🖻 I		findmax.cu:96 (0x9340b8)		•			
CUDA Thread (1,0,0) Block (0,0,0)			🕸 (162,0,		Warp 5 Lane 2 🖻 findm		ndmax.cu:96 (0x9340b	8)	0			
Block (0,0,0) [sm: 0] (256 Active Threads)			(163,	,0,0)),0) Warp 5 Lane 3 🖻 findmax.cu:96		ndmax.cu:96 (0x9340b	8)				
(1)	Reack (1 0 0) [cm· 2] (256 Active Threade)		164	0.0)	Maro 51	ane / ile fi	ndmax cu:06 (0x0340h					
ß) findmax.cu 🛙	-	° 🗆	🕬= Varia	bles 🛙	1	•• 🖻 💣 🗶 🔆	1 2				
			1	Name		Туре	T(0,0,0)B(0,0,0)	T(1,0,0)B(0,0,0)				
	uint32_t max = array[[]rstElementIndex]; uint32 t maxIndex = firstElementIndex:			🕨 🥭 arr	ay	@generic u	int32 0x400100000	0x400100000				
uint32_t nextElement;				(×)= ma	⇔= maxIndex uint32_		<value <value="" optimizer="" optir<="" td=""><td></td></value>					
<pre>uint32_t i = firstElementIndex + threadsCount;</pre>				⇔ i uint32_t		uint32_t	8192	8193				
<pre>for (; i < ARRAY SIZE; i += threadsCount) {</pre>				🕬 nextElement @register uint32			iint32 <mark>436811</mark>	7602589				
~	<pre>nextElement = array[i];</pre>			(×)• Firs	tElement	@register u	int32 <value optimize<="" td=""><td><value optimized<="" td=""><td></td></value></td></value>	<value optimized<="" td=""><td></td></value>				
~	max = nextElement:			(×)• ma	x	@register u	int32 <value optimize<="" th=""><th><value optimized<="" th=""><th></th></value></th></value>	<value optimized<="" th=""><th></th></value>				
	<pre>maxIndex = i;</pre>											
	}		J	📽 Expr	essions 🕅	3 1	🕫 🖻 🕂 💥	1 2 - 0				
	threadMax[threadIdx x] = max:			Express	sion	Туре	T(0,0,0)B(0,0,0)	T(1,0,0)B(0,0,0)				
threadMaxIdx[threadIdx.x] = max, threadMaxIdx[threadIdx.x] = maxIndex;				(x)• arr	ay[i] @generic uint32 436811		int32 436811	7602589				
				🖶 Add new exp								
	reduce(threadmax, threadmaxtdx),											
	<pre>if (!threadIdx.x) { // After reduce max will be in thread</pre>	2										
	array[blockIdx.x] = threadMax[0]; array[blockIdx.x + BLOCKS] = threadMaxIdx[0];											
	}		Ļ									
) Þ										
	□◆											
		_	_	_	_	1						
	<pre>} } supplementations + process = runsequertex[e];</pre>											
	array[blockIdx.x] = threadMax[0];	ML										
	<pre>if (ithreadIdx.x) { // After reduce max will be in thread</pre>	V										
	<pre>reduce(threadMax, threadMaxIdx);</pre>											

Nsight Debugger

- Seamless and simultaneous debugging of both CPU and GPU code
- View program variables across several CUDA threads
- Examine execution state and mapping of the kernels and GPUs
- View, navigate and filter to selectively track execution across threads
- Set breakpoints and single-step execution at both source-code and assembly levels
- Includes CUDA-MEMCHECK to help detect memory errors

NVIDIA[®] Nsight,^{*} Eclipse Edition



Nsight Profiler

۲

- Easily identify performance bottlenecks using a unified CPU and GPU trace of application activity
- Expert analysis system pin-points potential optimization opportunities
- Highlights potential performance problems at specific source-lines within application kernels
- Close integration with Nsight editor and builder for fast edit-build-profile optimization cycle
- Integrates with the new nvprof commandline profiler to enable visualization of profile data collected on headless compute nodes

CUDA on Mac!



NVIDIA GPUDirect[™] now supports RDMA



Server 1Server 2RDMA: Remote Direct Memory Access between any GPUs in your cluster

CUDA Compiler Contributed to Open Source LLVM

Developers want to build front-ends for Java, Python, R, DSLs

Target other processors like ARM, FPGA, GPUs, x86





Try out CUDA 5

CUDA 5.0 Release Candidate

- Available early next week!
- Full support for all CUDA 5.0 features
- Use GPU linking and NSIGHT EE—both work with Fermi & GK10x
- Peruse early documentation and header files for GK110 features
 - SM 3.5 support and Dynamic Parallelism
- Provide feedback to NVIDIA via CUDA Forums and <u>CUDA_RegDev@nvidia.com</u>
- CUDA 5.0 Preview (alpha)
 - Become a registered developer and download CUDA 5.0 preview <u>http://developer.nvidia.com/user/register</u>

How to get started

www.nvidia.com/cudazone

www.nvidia.com/getcuda

GTC 2013 | March 18-21 | San Jose, CA The Smartest People. The Best Ideas. The Biggest Opportunities.

Opportunities for Participation:

SPEAK - Showcase your work among the elite of graphics computing

Call for Sessions: August 2012 Call for Posters: October 2012

REGISTER - learn from the experts and network with your peers

- Use promo code GM10SIGG for a 10% discount

SPONSOR - Reach influential IT decision-makers



Learn more at www.gputechconf.com